

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

1 ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

магістр

(освітній ступінь (освітньо-кваліфікаційний рівень))

на тему: Методи та засоби реалізації програмних агентів для проектів
класу «розумне місто»

Виконав: студент VI курсу груп СНм-
и 61

спеціальності (напряму 122
підготовки)

Комп'ютерні науки

(шифр і назва спеціальності (напряму підготовки))

Забігайло О.І.
(підпис) (прізвище та ініціали)

Керівник Пасічник В.В.
(підпис) (прізвище та ініціали)

Нормоконтроль Мацюк О.В.
(підпис) (прізвище та ініціали)

Рецензент Михалик Д.М.
(підпис) (прізвище та ініціали)

м. Тернопіль – 2019

АНОТАЦІЯ

Методи та засоби реалізації програмних агентів для проектів класу «розумне місто» // Дипломна робота освітнього рівня "Магістр" // Забігайло Олександр Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНм-61 // Тернопіль, 2019 // С. , рис. – , табл. – , кресл. – , додат. – , бібліогр. – .

Ключові слова: АГЕНТ, АРХІТЕКТУРА, ВЗАЄМОДІЯ, ПРОГРАМА, СЕРЕДОВИЩЕ, СИСТЕМА, СТРУКТУРА, РОЗУМНЕ МІСТО.

Дипломна робота присвячена дослідженню методів та засобів реалізації програмних агентів для проектів класу «розумне місто». В першому розділі досліджено середовище сучасних проектів «розумних міст», проаналізовано існуючі програмні агенти в системах «розумного міста», досліджено та описано інструментарій для створення програмних агентів в «розумному» міському середовищі, розглянуто мови програмування і програмні платформи для створення «розумних» програмних агентів.

В другому розділі дипломної роботи досліджено засоби специфікації моделей «розумних» програмних агентів, проаналізовано інструментальні засоби для розроблення та побудови «розумних» програмних агентів, наведено порівняльний аналіз інструментальних середовищ.

В третьому розділі дипломної роботи, запропонована архітектура мультиагентних застосунків «розумного міста», розроблена модель ресурсів «розумного» програмного агента, наведено опис розробленого програмного модуля «розумного» міського програмного агента. Виконано розділи «Спеціальна частина», «Обґрунтування економічної ефективності», «Охорона праці та безпека в надзвичайних ситуаціях», «Екологія».

ANNOTATION

Methods and tools of software agents implementation for the “smart city”-class projects // Master's Thesis // Alexander Zabigailo // Ivan Puliuy Ternopil National Technical University, Department of Computer-Information Systems and Software Engineering, Department of Computer Sciences, SNm-61 group // Ternopil, 2019 // P. , fig. - , Table. - , chair - , add. - , bibliography - .

The diploma thesis is devoted to the research of methods and means of implementation of software agents for projects of the "smart city" class. The first section explores the environment of modern "smart cities" projects, analyzes existing software agents in "smart city" systems, investigates and describes tools for creating software agents in "smart" urban environments, discusses programming languages and software platforms for creating "smart" software agents.

The second section of the thesis investigates the means of specification of models of "smart" software agents, analyzes the tools for the design and construction of "smart" software agents, provides a comparative analysis of tool environments.

In the third section of the thesis, the architecture of multi-agent applications of "smart city", the model of resources of "smart" software agent, the description of the developed software module "smart" urban software agent. The sections “Special part”, “Justification of economic efficiency”, “Occupational health and safety”, “Ecology” have been completed.

Keywords: AGENT, ARCHITECTURE, INTERACTION, PROGRAM, ENVIRONMENT, SYSTEM, STRUCTURE, SMART CITY.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ADF (англ. Agent Definition File) – Файл ініціалізації агента.

ACL (англ. Agent Communication Language) – Мова комунікації агентів.

AMS (англ. Agent Management System) – Система управління агентами.

BAE (англ. Building Agent Environment) – Середовище побудови агентів.

BDI (англ. Belief-Desire-Intention) – «Віра»-«Бажання»-«Намір».

CLDC (англ. Connected Limited Device Configuration) – Підключення пристрою з обмеженою конфігурацією.

DAF (англ. Distributed Agent Platform) – Розподілена агентна платформа.

FD (англ. Directory Facilitator) – Служба каталогів.

JADE (англ. Java Agent Development Framework) – Java-фреймворк розроблення агентів.

JVM (англ. Java Virtual Machine) – Віртуальна машина Java.

KAoS (англ. Knowledgeable Agent-oriented System) – Знаннєва агент-орієнтована система.

KIF (англ. Knowledge Interchange Format) – Формат обміну знаннями.

KOML (англ. Knowledge Query and Manipulation Language) – Запит на знання та мова маніпуляції.

LEAP (англ. Lightweight Extensible Agent Platform) – Легка платформа розширюваного агента.

MDS (англ. Multiple Domains Support) – Підтримування багатьох доменів.

PAL (англ. Project Accessories Library) – Бібліотеки класів проекту.

RAMM (англ. Reticular Agent Mental Model) – Ментальна модель ретикулярного агента.

RADL (англ. Reticular Agent Definition Language) – Мова ініціалізації ретикулярних агентів.

RRТАЕ (англ. Reticular's Run-Time Agent Engine) – Рушій опрацювання ретикулярних агентів в режимі реального часу.

SKTP (англ. Simple Knowledge Transfer Protocol) – Простий протокол передачі знань.

SL (англ. Semantics Language) – Семантична мова.

TCL (англ. Tool Command Language) – Інструменти командної мови.

ІЗ – Інструментальні засоби.

БЗ – База знань.

МАС – Мультиагентна система.

ПК – Персональний комп'ютер.

ШІ – Штучний інтелект.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.1 Середовище сучасних проектів «розумних міст»	12
1.2 Аналіз існуючих програмних агентів в системах «розумного міста»	14
1.3 Опис інструментарію для створення програмних агентів в «розумному» міському середовищі	17
1.4 Мови програмування і програмні платформи для створення «розумних» програмних агентів	23
1.5 Висновок до першого розділу	29
2 ЗАСОБИ СПЕЦИФІКАЦІЙ ТА РОЗРОБЛЕННЯ «РОЗУМНИХ» МІСЬКИХ ПРОГРАМНИХ АГЕНТІВ.....	31
2.1 Засоби специфікації моделей «розумних» програмних агентів	31
2.2 Інструментальні засоби для розроблення та побудови «розумних» програмних агентів	34
2.2.1 Середовище реалізації «розумних» програмних агентів JADE ..	35
2.2.2 Середовище реалізації «розумних» програмних агентів JACK..	40
2.2.3 Середовище AgentBuilder.....	44
2.2.4 Середовище Bee-gent	46
2.3 Порівняльний аналіз інструментальних середовищ	49
2.4 Висновок до другого розділу	52
3 АРХІТЕКТУРА ТА МОДЕЛІ ПРОГРАМНИХ АГЕНТІВ У «РОЗУМНОМУ МІСТІ».....	53
3.1 Архітектура мультиагентних застосунків «розумного міста»	53
3.2 Модель ресурсів «розумного» програмного агента	55
3.3 Опис розробленого програмного модуля «розумного» агента	58
3.4 Висновок до третього розділу	60

4 СПЕЦІАЛЬНА ЧАСТИНА	62
4.1 Реалізація моделі міського «розумного» програмного агента	62
4.2 Висновок.....	70
5 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ	71
5.1 Розрахунок норм часу на виконання науково-дослідної роботи.....	71
5.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи.....	72
5.3 Розрахунок матеріальних витрат.....	76
5.4 Розрахунок витрат на електроенергію	77
5.5 Розрахунок суми амортизаційних відрахувань	77
5.6 Обчислення накладних витрат.....	78
5.7 Складання кошторису витрат та визначення собівартості науково-дослідницької роботи	79
5.8 Розрахунок ціни проведених науково-дослідних робіт.....	80
5.9 Визначення економічної ефективності і терміну окупності капітальних вкладень.....	81
5.10 Висновок	82
6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	84
6.1 Покращення управління охорони праці в організації шляхом впровадження інформаційних технологій.....	84
6.2 Організаційно-технічні заходи та елементи системи протипожежного захисту робочих місць користувачів ПК.....	87
6.3 Забезпечення електробезпеки користувачів ПК.....	89
6.4 Оцінка дії радіоактивного забруднення місцевості після ядерного вибуху на виробничу діяльність муніципальних підприємств, установ та організацій	92
6.5 Висновок.....	97
7 ЕКОЛОГІЯ	99
7.1 Екологічна ситуація в Україні	99

7.2 Вимоги до моніторів (ВДТ) і ПЕОМ	103
7.3 Висновки до розділу	105
ВИСНОВКИ.....	106
ПЕРЕЛІК ДЖЕРЕЛ.....	108
ДОДАТКИ	

ВСТУП

Актуальність теми. На даний час невідомо зростає популярність проектів класу «розумне місто», спричиняючи зростанням обсягів міських даних та виникає необхідність у пошуках покращених та вдосконалених методів доступу та обробки, представлення та систематизації інформації.

Розвиток інформаційних та телекомунікаційних технологій і їх активне впровадження в міське життя сприяє збільшенню обсягів інформації, необхідної для роботи міських інформаційних систем інтегрованих в інноваційні проекти «розумних міст». Обсяг цих даних постійно збільшується, зростаючи за рахунок деталізації у вже присутніх інформаційних джерелах інтегрованих в міське середовище та за рахунок появи принципово нових джерел.

На даний час в міське середовище активно інтегруються та впроваджуються стек інформаційних технологій розроблених з використанням мультиагентних систем, протоколів, програмних застосунків, системних архітектур та реалізацій.

Тому актуальною є науково-практична задача розробки методів побудови та використання «розумних» мультиагентних систем в проектах класу «Розумне місто» як основи міських інформаційних систем, а також розробки підходу до використання інформаційних технологій побудови та використання «розумних» програмних агентів інтегрованих в міське середовище з метою пошуку та видобування нових даних та знань.

Мета і задачі дослідження. Покращення поінформованості жителів та гостей міста про перебіг процесів у міському середовищі за рахунок впровадження «розумних» мультиагентних програмних рішень.

Поставлена мета дослідження дозволила сформулювати перелік задач:

- дослідити середовище інноваційних проектів «розумних міст»;

- проаналізувати існуючі програмні агенти в системах «розумного міста»;
- дослідити інструментарій для створення програмних агентів в «розумному» міському середовищі;
- дослідити мови програмування і програмні платформи для створення «розумних» програмних агентів.
- дослідити засоби специфікації моделей «розумних» програмних агентів;
- проаналізувати інструментальні засоби для розроблення та побудови «розумних» програмних агентів;
- виконати та подати порівняльний аналіз інструментальних середовищ.
- розробити архітектуру мультиагентних застосунків «розумного міста»;
- розробити модель ресурсів «розумного» програмного агента.

Об'єкт дослідження – «розумний» програмний агент для подання навчальних даних та знань жителям та гостям міста.

Предмет дослідження. Методи та засоби організації «розумних» мультиагентних застосунків та систем в міському середовищі.

Методи дослідження. Методи аналітичного опрацювання наукових публікацій та джерел. Методи системного аналізу. Методи кросплатформного програмування. Методи системного та логічного програмування.

Наукова новизна одержаних результатів. На основі проведеного аналізу засобів специфікації моделей «розумних» програмних агентів та інструментальних засобів для їх розроблення та побудови було розроблено модель ресурсів «розумного» програмного агента реалізовану на основі запропонованої архітектури мультиагентних застосунків «розумного міста». Що на відміну від існуючих моделей та систем враховує особливості

формування мультиагентних застосунків в умовах гіперскладних систем «розумного міста».

Практичне значення одержаних результатів. Розроблено прототип програмного модуля «розумного» агента для використання в проектах класу «розумне місто».

Апробація результатів магістерської роботи проведена на двох наукових конференціях з публікацією тез доповідей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Середовище сучасних проектів «розумних міст»

Міста завжди стикаються з потребами надання послуг громадянам для підвищення якості життя та покращення сервісу. Хоча міста розвиваються в міру доступних можливостей та ресурсів, це спричиняє багато проблем, що впливають на навколишнє середовище та сьогодення [1]. Технології завжди були центром людської уваги і протягом тривалого часу сильно змінювали наш світ і життя. Цифрові дані пов'язані з фізичними об'єктами, людьми та пристроями впливають на нашу працю, подорожі, спілкування та взаємодію з оточенням змінюючи різні сфери міського буття, зокрема, охорону здоров'я, моніторинг навколишнього середовища, міські системи та управління, тощо.

«Розумні» міські ініціативи вивчають та використовують досягнення сучасних інформаційних та комунікаційних технологій, таких як Інтернет-пристроїв («IoT») [2], великих за обсягом даних («BigData») [3], хмарних обчислень («Cloud Computing») [4], Повсюдних мереж («Ubiquitous Network») [5], мобільних технологій («Mobile Technology») [6] для вирішення міських задач, таких як зменшення споживання енергії, уникнення транспортних заторів та зменшення забруднення навколишнього середовища. Проте поточні служби все ще значною мірою обмежуються відособленими доменами. Незважаючи на багатогранність доступних міських інформаційних наборів агрегованих з численних джерел даних, міська влада все ще стикається з низкою труднощів у впровадженні, підтримці та оптимізації операцій та взаємодії між різними міськими департаментами та службами [7]. Все ще залишається не вирішеною потреба в розумних міських застосуваннях, які підтримують сталий розвиток муніципальних програм. Технічні проблеми включають неоднорідність, швидкість, різноманітну якість, невизначеність та неповноту даних, зібраних з міського середовища.

Потребують напрацювання програмні засоби, які можуть забезпечити безперервне та динамічне спостереження міського середовища з метою інформування муніципальної адміністрації та користувачів про міські події, їх час та локалізацію, вплив на громадян, туристів, компанії та адміністрацію міста. [8] Для формування подібних різнопланових уявлень про поточний стан міста потребують розробки комплексні системи, що дозволять інтегрувати, маніпулювати та обробляти величезними та різноманітними наборами даних гнучким і розширюваним способом. Такі системи потребують розробки та напрацювання потужного програмно-алгоритмічного апарату аналітичного опрацювання даних. Складні системи класу «Розумне місто» потребують проектування та реалізації:

- Модулів сенсорики та агрегації даних на основі алгоритмів адаптивних до змін у джерелах вхідних даних та розроблених для мінімізації інформаційних втрат.

- Модулів виявлення подій, які генерують семантично анотовану інформацію вищого рівня.

- Модулі консолідації даних, які реалізують алгоритми з можливістю автоматичного пошуку придатних джерел даних під час виконання, відповідно до специфікацій інформаційних потреб користувача.

- Модулі моніторингу якості вхідних даних із застосуванням алгоритмів машинного навчання для оцінки їх якості.

- Модулі контексної фільтрації з можливостями постійного моніторингу поточної активності користувачів для автоматичного вибору відповідних подій.

- Модулі підтримки прийняття рішень, які поєднують семантичні технології та рекомендаційні системи для забезпечення масштабованої підтримки управлінських та користувацьких рішень.

Присутня потреба концептуалізації прототипу з використанням «розумних» програмних агентів для динамічної інтеграції інформації про

витрати послуг в ресурсних та соціокомунікаційних мережах міста, зокрема, води, газу, тепла та електроенергії, для формування стратегії надання послуг та ціноутворення в конкурентному середовищі «розумного міста», та реалізації рекомендаційної системи, щодо вибору оптимальних режимів використання муніципальних послуг міськими жителями. [9]

1.2 Аналіз існуючих програмних агентів в системах «розумного міста»

Відповідно до класичного визначення, «розумний» програмний агент – це програма-посередник, яка взаємодіє з користувачами або іншими програмними засобами в інформаційних системах «розумного міста» з метою виконання певних дій від імені користувачів, або інших програм. [10]

Можна дати інше відносно вільно означення «розумного» програмного агента, який виступає в якості самодостатньої програми, здатної контролювати свої власні рішення та дії, опираючись на своє власне сприйняття оточення, та спрямована на реалізацію однієї або декількох цілей. Якщо розглядати відмінності між «розумним» програмним агентом і застосунком, то можна сказати, що головна відмінність полягає в поведінці агента. [11]

Концепція «розумного» агента забезпечує зручний і потужний спосіб опису складної програмної сутності, яка здатна діяти з певним ступенем автономності з метою виконання завдань від імені користувача. «Розумний» агент визначається за допомогою опису його поведінки. [12]

Існує кілька підходів до визначення поняття «розумного» програмного агента, проте узагальнена їх суть полягає в тому, що «розумні» агенти більш автономні, ніж об'єкти. «Розумний» агент має автономністю в плані вибору завдань і пріоритетів. Також «розумному» агенту властива гнучка модель поведінки. У завдання «розумного» програмного агента входять:

1. Самостійна робота і контроль власних дій.
2. Взаємодія з іншими «розумними» агентами.
3. Зміна поведінки в залежності від зміни стану навколишнього середовища.
4. Видача адекватної та максимально-повної інформації щодо виконання заданої функції і т.п.

«Розумні» мультиагентні системи в проектах класу «розумне місто» – це системи, що складаються з автономних «розумних» агентів, що взаємодіють один з одним і пасивного навколишнього середовища, в якому агенти існують і на яке також можуть безпосередньо впливати. [13]

Існують різні класифікації «розумних» програмних агентів. Розглянемо найбільш популярну класифікацію, в якій виділяють чотири основні типи програмних агентів [14]:

1. «Розумні» агенти-покупці або торгові боти.
2. Користувальницькі та персональні «розумні» агенти.
3. «Розумні» агенти моніторингу і спостереження.
4. «Розумні» агенти для видобування та аналітичного опрацювання даних.

Ця класифікація не зовсім актуальна на даний час. Так наприклад, за класифікацією «розумні» агенти-покупці переглядають мережеві ресурси з метою отримання інформації щодо товарів і послуг. Але на сьогоднішній момент індексуючі роботи в своєму розвитку пішли набагато далі і виходять далеко за рамки просто пошукових систем. [15]

Прикладом сьогоднішнього «розумного» торгового бота виступає інструмент трейдера, він являє собою спеціальне високотехнологічне програмне забезпечення, яке автоматично здійснює торгівлю на біржі, без допомоги людини. «Розумний» торгівельний бот працює за певними правилами та за своєю наперед визначеною стратегією. Стратегій може існувати багато і вони можуть суттєво відрізнятись. «Розумні» боти також

можуть працювати на комп'ютерах користувачів, в «хмарах» або у формі плагінів для торгових систем.

«Розумні» боти можуть вирішувати кілька основних завдань. Перше з них – це автоматизація рутинних операцій. Деякі стратегії торгівлі потребують постійного виконання ордерів та облікування великих обсягів даних, що досить важко робити вручну. Інші «розумні» стратегії потребують дуже швидкого реагування на зміни навколишнього середовища і прийняття простих, але швидких та ефективних рішень. «Розумні» боти також можуть використовуватися працівником муніципалітету як допоміжний інструмент для будь-яких операцій.

Призначені для користувача «розумного міста» або персональні «розумні» агенти по даній класифікації – це «розумні» агенти, які діють від імені користувача. Прикладом таких дій є відправка даних, отримання та автоматичне опрацювання даних. Є декілька версій таких «розумних» агентів, в тому числі «newshub» і «CNN».

Персональні «розумні» агенти можуть виконати ряд наступних дій:

1. Здійснювати пошук інформації згідно заданої теми та запиту.
2. Автоматичне заповнення веб-форм та зберігання інформації для використання в майбутньому.
3. Сканування веб-сторінок для пошуку і виділення тексту, який являє собою «важливу» частину інформації.
4. Надають можливість «обговорювати» різні теми.
5. Сприяти онлайн пошуку шляхом сканування відомих дощок оголошень і відправки даних, які відповідають бажаним критеріям.
6. Профілювати синхронізацію різнорідних соціальних мереж.

«Розумні» програмні агенти з моніторингу та спостереження використовуються для спостереження за об'єктами в середовищі «розумного міста» і передачі інформації на обладнання, як правило, на комп'ютерні системи. «Розумні» агенти можуть відстежувати рівень запасів матеріалів

компанії, стежити за цінами конкурентів і доводити їх до відома компанії, аналізувати маніпуляції за інсайдерською інформацією і чутками, і т.д. [16]

Наприклад, в лабораторії реактивного руху НАСА є «розумний» агент, який відстежує наявність і стан обладнання, планує замовлення на придбання нового обладнання з урахуванням оптимізації витрат, а також стежить за наявністю продуктів харчування. Такі «розумні» агенти зазвичай здійснюють моніторинг складних комп'ютерних мереж в середовищі «розумного міста» і можуть стежити за зміною кожного комп'ютера, підключеного до мережі. [17]

Особливими «розумними» агентами з моніторингу та спостереження є структури «розумних» агентів, які використовуються для моделювання процесів прийняття рішень під час тактичних операцій в міському середовищі. «Розумні» агенти здійснюють моніторинг стану активів і отримують цілі від «розумних» агентів верхнього рівня. «Розумні» агенти домагаються досягнення цілей з наявними активами, зводячи до мінімуму витрати активів при максимальному досягненні мети. [18]

В сучасних умовах при розгляді даної класифікації «розумні» програмні агенти з видобутку та аналізу даних необхідно віднести до «розумних» агентів покупців або торгівельних ботів.

1.3 Опис інструментарію для створення програмних агентів в «розумному» міському середовищі

Для створення «розумних» програмних агентів в «розумному місті» використовують мовні описи та реалізацію, які зручно представляти у вигляді багатоваріаційної структури, котра включає п'ять шарів [19]:

1. Мовні та програмні засоби реалізації «розумних» агентів.
2. Мови комунікації «розумних» агентів.
3. Мови опису поведінки «розумних» агентів та законів середовища.

4. Мови представлення та управління знаннями.

5. Мови формалізації і специфікації «розумних» агентів і «розумних» мультиагентних систем (МАС).

До засобів взаємодії «розумних» агентів відносяться мови комунікації агентів (FIPA ACL, KQML). Мова комунікації агентів забезпечує обмін знаннями та інформацією між агентами. FIPA ACL на відміну від таких засобів як RPC, RMI, CORBA, що забезпечують обмін інформацією між застосунками, має більш складну семантику і володіє наступними перевагами [20]:

- FIPA ACL управляє судженнями, правилами, діями, а не семантично не зв'язаними об'єктами;
- повідомлення FIPA ACL описує очікуваний стан, а не процедуру чи метод.

Однак ACL не охоплює повний спектр об'єктів, які можуть бути помічені «розумними» агентами, наприклад, планами, цілями, досвідом, стратегіями. На технічному рівні, використовуючи ACL, «розумні» агенти транспортують повідомлення по мережах, використовуючи протоколи низького рівня, наприклад SMTP, TCP / IP, POP3 або HTTP.

Мова взаємодії «розумних» агентів (ACL) повинна передавати будь-яку інформацію між різними «розумними» програмними агентами. Наявні два підходи до проектування мови взаємодії «розумних» агентів:

1. Процедурний, включає в себе дотримання процедурних директив/команд. Це може бути реалізовано за допомогою таких програмних середовищ, як Java або Tcl.

2. Декларативний – зв'язок базується на декларативних інструкціях, типу визначень, припущень, знань і т.п.

Через обмеження на процедурні підходи (наприклад, такі сценарії важко координувати, об'єднати), декларативні мови набули перевагу при

створенні мов взаємодії «розумних» агентів. Одні з найбільш популярних декларативних мов – KQML зі своїми діалектами і FIPA ACL.

У контексті практичної побудови «розумних» агентів і МАС головну роль відіграють інструментальні засоби програмування і комунікації «розумних» агентів. Мови комунікації (ACL, KQML) і координації «розумних» агентів (AgenTalk) забезпечують узгоджене взаємодії «розумних» агентів – циркуляцію інформації, передачу запитів послуг, реалізують механізми діалогів, підтримують співпрацю між «розумними» агентами, спрямовану на досягнення спільної мети і, як наслідок, формування колективів «розумних» агентів. Ці мови можна розглядати як багаторівневі структури, що включають рівень представлення знань, рівень переговорів або координації, рівень стратегій комунікації, і т. п. Так мова KQML, використовується для підтримки взаємодії «розумних» агентів в розподілених застосунках та спирається на спеціальний протокол перенесення знань SKTP. У свою чергу, мова ACL, яка претендує на роль стандарту для комунікації «розумних» агентів, складається з трьох частин – словника, "внутрішнього мови" KIF і "зовнішньої мови" KQML. Повідомлення, передане на мові ACL, може трактуватися як KQML-вираз, «аргументами» якого виступають пропозиції в форматі KIF, побудовані з елементів словника ACL.

Програмні засоби, застосовані для реалізації «розумних» агентів, діляться на три основні класи:

1. Мови програмування.
2. Бібліотеки розробки агентів.
3. Середовища розробки агентів.

При програмуванні агентів найчастіше застосовуються мови об'єктно-орієнтованого програмування (C++, Java), рідше використовуються символічні мови і мови логічного програмування (LISP, Oz).

Серед існуючих бібліотек «розумних» агентів можна виділити «Intelligent Agent Library» – комерційний продукт фірми «Bits&Pixels», систему «Кайса», розроблену фірмою «Fujitsu» і «Agentx» – мережа високоефективних бібліотек розподілених обчислень в програмному середовищі Java, запропоновану фірмою «International Knowledge Systems». Бібліотека «розумних» агентів «Intelligent Agent Library» являє собою набір засобів, призначених для забезпечення комунікації «розумних» агентів і побудови їх груп. Вона побудована на основі мови KQML і містить ілюстративні приклади «розумних» агентів, що функціонують у веб-застосунках. Ця бібліотека підтримує розробку мобільних «розумних» агентів.

Одним із найвідоміших та поширених інтегрованих середовищ для розробки «розумних» програмних агентів є «AgentBuilder» фірми «Reticular Systems, Inc». Це засіб складається з двох основних компонентів: інструментарію (Toolkit) і виконавчої системи. Інструментарій включає:

1. Засоби управління процесом розробки «розумного» агентно-базованого програмного забезпечення.
2. Засоби аналізу діяльності функціонуючих «розумних» агентів;
3. Засоби проектування та розробки мереж взаємодіючих «розумних» агентів;
4. Засоби моделювання поведінки окремих «розумних» агентів;
5. Засоби відлагодження та тестування «розумних» програмних агентів.

Виконавча система містить «розумну» машину агента («agent engine»), яка формує середовище для реалізації «розумних» програмних агентів. «Розумні» агенти, розроблені за допомогою середовища «AgentBuilder», взаємодіють на мові KQML, в основі якого лежать примітивні дії – перформативи. [21]

Призначення інструментальних засобів (ІЗ) «AgentBuilder» полягає в тому, щоб надати розробнику «розумного» агентно-базованого програмного

забезпечення, інтегроване середовище, що дозволяє швидко і легко створювати «розумні» агенти і складні програми на їх основі. «Розумні» агенти, створені за допомогою «AgentBuilder», можуть бути реалізовані на будь-якій віртуальній Java-машині. У таблиці 1.1 подані приклади засобів для розробки «розумних» агентів.

Таблиця 1.1 – Засоби для розроблення «розумних» агентів

Назва мови (засоби розробки)	Коротка характеристика/призначення
AgentTalk	Засоби для розробки автономних «розумних» мобільних агентів на базовій мові сценаріїв Tcl з підтримкою Tk-інструментарію для створення графічного інтерфейсу.
AgentTool	«Розумна» агентна структура на базі Java. Використання графічного інтерфейсу для конструювання систем.
FIPA-OS	Реалізація елементів, що містяться в FIPA-специфікації та взаємодії «розумних» агентів. Java-реалізація архітектури для розробників «розумних» агентів.
Echelon	Засоби для розробки «розумних» агентів у мережах LonWorks.
JAFMAS	Програма, що забезпечує створення мультиагентної системи на Java. Підтримує міжагентну взаємодію засобами KQML.
Remembrance Agents	Засоби для розробників «розумних» агентів, спостереження за користувачами та пропонуючими необхідну на даний час інформацію.
Ummon	Середовище для створення самонавчаючихся «розумних» агентів. Містить методи III для забезпечення взаємодії.
Virtual Secretary Project (ViSe)	Створення моделей користувача на основі інтерактивних «розумних» агентів для виконання секретарських завдань (технологія Tcl/TclX/Tix/Tk)

У таблиці 1.2 подано порівняння параметрів окремих інструментальних засобів, визначених у таблиці 1.1, які використовуються для розроблення «розумних» агентів.

Таблиця 1.2 – Порівняння параметрів інструментальних засобів для розроблення «розумних» агентів

Параметр	Назва інструментального засобу					
	JAFMAS	Agen Talk	Agent Tcl	Telescript	Swarm	Echelon
Мова розробки	Java	Lisp	Tcl	Telescript	Objective C	Silicon chip
Підтримка ОС	Vci	UNIX	UNIX	UNIX	UNIX	Echlon chip
Реалізація агента	програмна	програмна	програмна	програмна	програмна	апаратна
Об'єктно-орієнтоване програмування	+	-	-	+	+	+
Наявність у агента свого плану дій	+	-	+	+	+	+
Комунікаційний протокол	TCP/IP і UDP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP	TCP/IP
Мобільність агентів	+	-	-	-	-	-

Аналіз табл. 1.1 і 1.2 показав, що засоби для розроблення «розумних» агентів можна умовно розділити на дві групи: засоби, побудовані на базі мови Java, і засоби, побудовані на базі інших мов. Засоби першої групи призначені для розробки мережевих додатків на базі «розумних» мобільних агентів, що взаємодіють через протокол TCP/IP.

1.4 Мови програмування і програмні платформи для створення «розумних» програмних агентів

Для програмування «розумних» агентів можуть застосовуватися: універсальні мови («Java», «C ++», «Visual Basic» та ін.), мови представлення знань («SL», «KIF»), мови переговорів та обміну знаннями («KQML», «AgentSpeak», «April»), мови сценаріїв («Tcl/Tk», «Python», «Perl5» і ін.), спеціалізовані мови («TeleScript», «COOL», «Agent0», «AgentK» і ін.), символічні мови і мови логічного програмування («Oz», «ConGolog», «IMPACT», «Dylog», «Concurrent METATEM», «Ehhf» і ін.), а також інші мови і засоби розробки «розумних» агентів. На рисунку 1.1 подана класифікація мов міжагентних комунікацій. [22]

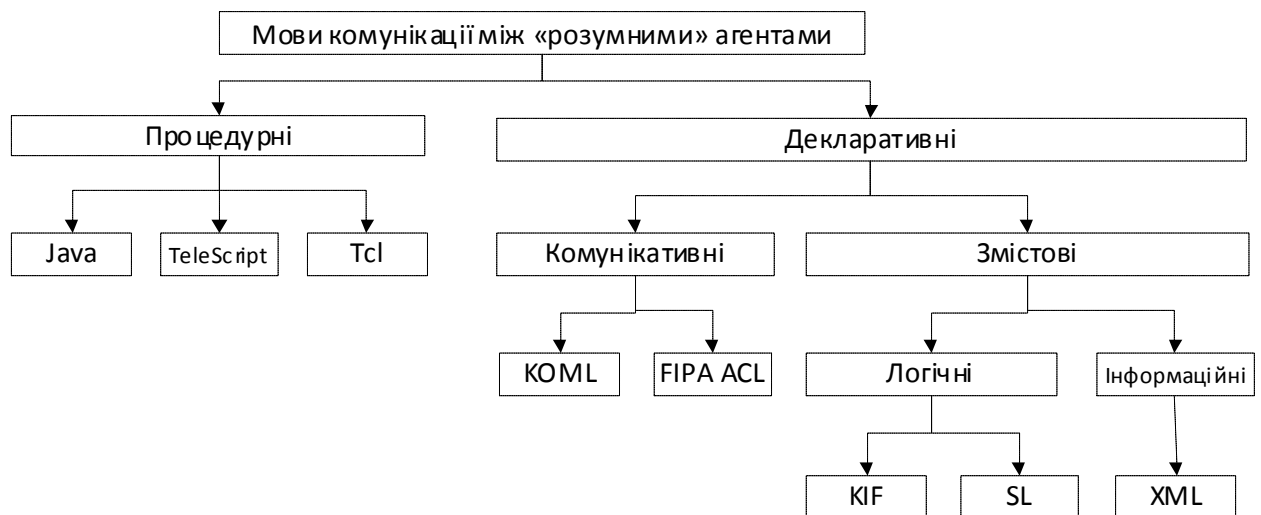


Рисунок 1.1 – Класифікація мов «розумної» міжагентних комунікацій

Завдяки тому, що мова програмування Java є архітектурно-незалежною та компілює код в машинно-незалежні застосунки, розроблені за допомогою засобів цієї групи, можуть працювати на багатьох різних інформаційно-технологічних платформах. Різниця між засобами в тому, що не всі надають підтримку мовного обміну повідомленнями за допомогою

«KQML». Недоліком є відсутність класів для визначення групової або «соціальної» поведінки «розумних» агентів. [23]

Інструментальні засоби другої групи, в основному, призначені для проектування складних динамічних «розумних» агентних структур і реалізації середовищ, хоча є окремі екземпляри для створення мобільних застосунків («Telescript», «Agent Tcl»). При цьому «розумні» агенти розробляються згідно BDI-моделі. Комунікації здійснюються з використанням протоколу TCP/IP. Однак такі засоби мають обмежені можливості узгодження і кооперації «розумних» агентів та працюють на обмеженій кількості платформ, оскільки мови програмування не уніфіковані.

Більшість комунікативних «розумних» агентних мов ACL побудові на основі мовної взаємодії (мовні дії виражаються за допомогою стандартних ключових слів).

Відомі два підходи до розробки таких мов – процедурний і декларативний. При першому підході комунікації відбуваються під час виконання інструкцій. Мова проектується за допомогою Java або TCL. При декларативному підході комунікації реалізуються на основі описів. Декларативний підхід отримав велике поширення для створення мов взаємодії «розумних» агентів, найбільш популярним з яких є KQML-структурована мова взаємодії «розумних» агентів. Ця мова використовується як мова взаємодії в різних «розумних» багатоагентних системах і середовищах для їх програмування, таких як «Agent-K», «LALO», «Java (tm) Agent Template (JATLite)». Цілі, аналогічні «KQML» має «KAoS».

При створенні «розумного» міжагентного повідомлення необхідна мова для подання самого змісту. Зазвичай для цього використовують «логічні мови», що представляють знання у вигляді логічних виразів (використовуючи LISP- подібний синтаксис), і «інформаційні мови», що встановлюють правила для опису типів інформаційних елементів. Типова логічна мова змісту – це мова «KIF», яка полегшує обмін знаннями між

системами III. Вона використовується разом з «KQML» в американських науково-дослідних проектах для подання знань в мультиагентних системах. Її синтаксис базується на «Common LISP». Формат KIF є декларативною мовою, яка дозволяє різним системам обмінюватися онтологіями (узгодження між різними системами щодо структур подання знань різних предметних областей), використовуючи тим самим в роботі обчислювальні переваги цих систем.

Друга логічна мова змісту – SL, запропонована FIPA. SL-пропозиції виражаються логікою ментальних відносин і дій. Ментальна модель «розумного» агента заснована на уявленні трьох примітивів: переконання, невизначеність, вибір. Основна властивість SL-логіки дозволяє змодельованим «розумним» агентам знаходитись у відповідності з їх ментальними відношеннями.

На відміну від мов «KIF» і «SL», мова «XML» не представляє інформацію у вигляді логічних виразів, а використовує інші типи структур. «розумні» XML-агенти мають здатність за запитом надавати інформацію з довільних джерел даних.

Для програмної реалізації «розумних» мультиагентних систем в проектах класу «розумне місто» використовуються спеціалізовані програмні системи, які містять набір засобів для програмного опису діяльності «розумних» агентів і стану середовища, контролю та управління процесом їх взаємодії і роботи. Такі системи називаються «розумними» агентними платформами проектів класу «розумне місто».

Існує досить великий набір платформ, придатних для створення «розумних» мультиагентних систем, і цей набір постійно поповнюється. У нього входять «NetLogo», «StarLogo», «Repast Simphony», «Eclipse AMP», «JADE», «Jason». Ці платформи реалізовані досить по-різному: від окремих середовищ розробки до вбудованих плагінів і підключених бібліотек. Вони можуть використовувати існуючі мови парадигм та мови, спеціально

розроблені для побудови «розумних» програмних агентів, наприклад, «AgentSpeak» в системі розробки «Jason».

За розмірами інструментарію платформи можна розділити на прості і складні. Прості («NetLogo», «StarLogo») мають маленький, але потужний інструментарій, що дозволяє швидко розробляти досить складні програми, однак при написанні великої системи цього інструментарію може не вистачити. Можливість його розширити та доповнити власними розробками, як правило, не надається. Тому, якщо для формалізації і налаштування моделі не підходять прості системи, то для реалізації краще вибрати складну, яка надає більше можливостей, хоча ними і важче користуватися. Розглянемо платформи для розроблення «розумних» агентів.

Платформа розробки Java-агентів «JADE» використовується для створення «розумних» мультиагентних систем і застосунків відповідно до стандартів «FIPA» для інтелектуальних агентів. Вона включає в себе середовище виконання агентів. Агенти реєструються і працюють під управлінням середовища в якому передбачені механізми створення та знищення, взаємодії та пошуку «розумних» агентів (див. рисунок 1.2). [24]



Рисунок 1.2 – Основні компоненти середовища JADE

Середовище включає систему управління «розумними» агентами AMS, яка надає засоби іменування та управління життєвим циклом «розумного» агента і службу каталогів – DF, що включає реєстраційний каталог. Використовуючи цей каталог, «розумний» агент може зв'язатися з іншими «розумними» агентами, необхідними йому для досягнення мети. Крім того, JADE містить бібліотеку класів, використовуваних для розробки «розумних» агентів і систем, включаючи набір графічних утиліт для адміністрування і спостереження за життєвим циклом «розумних» агентів, що функціонують в системі. Важливою особливістю JADE є підтримка виконання «розумних» агентів на пристроях з обмеженими можливостями («CLDC»), наприклад мобільних телефонах, мікроконтролерах і ін., За рахунок використання розширення LEAP.

Розглянемо середовище розробки «NetLogo». Дана система є типовим прикладом автономного середовища розробки «розумних» мультиагентних систем. До основних плюсів даної платформи відноситься великий обсяг засобів, що надаються для взаємодії з користувачем – різних кнопок і регуляторів для введення і редагування інформації, а також вільне розповсюдження системи.

Крім переліченого, дана платформа надає зі стандартною збіркою «NetLogo» дуже широкий набір прикладів моделей і систем. Подані приклади дійсно допомагають краще зрозуміти будову та можливості зазначеної «розумної» агентної платформи. Однак за багатством наданих інструментів прихована обмеженість технічних можливостей системи. По-перше, набір послуг, засобів візуалізації, хоч і широкий, але строго фіксований і регламентований, крім того, неможливо додавання будь-яких власних прийомів та засобів. По-друге, система позиціонується як засіб розробки програм, фактично примушуючи програміста писати всю програму одним файлом, без будь-якої чіпкої структури. Крім того, дуже специфічна мова

програмування, заснована на навчальній мові «Logo», котра містить численні методи, призначені спеціально для моделювання.

Мова програмування «NetLogo» є скриптовою мовою, тому написані на ній програми виконуються досить повільно. В даний час ведуться роботи щодо створення транслятора в байт-код, що обіцяє в перспективі істотне зменшення часу виконання.

Система не надає інструменти для розподіленого, незалежного виконання. Функціонуючі в ній суб'єкти позбавлені структури «розумного» агента, і є лише ідентифікаторами власного стану. Поведінкова логіка реалізується виконанням функцій на класі об'єктів, тобто, по суті, в рамках об'єктної парадигми програмування. З іншого боку, така простота взаємодії знаходиться в гармонії зі слабкими можливостями структурування програми та організації процесу розробки. Саме вона дозволяє писати невеликі не засмічені зайвими і практично непотрібними (хоча і важливими теоретично і структурно) елементами, але досить потужні програми. Цьому також сприяє і зазначена вище простота роботи з широким, хоча і жорстко обмеженим набором засобів візуалізації.

Розглянемо платформи з окремими модулями для розробки «розумних» мультиагентних систем. Окремі модулі являють собою бібліотеки, набори виключно розробницького засобів для написання систем, оскільки вони надають широкий набір методів взаємодії з користувачем. Однак вони вже не є фактично автономними системами, а втілюються у вигляді модулів до існуючих засобів розробки. Це досить добре, оскільки дозволяє поєднувати багатство стандартних засобів розробки зі спеціалізованими, призначеними виключно для «розумного» агентного моделювання засобами. До даного класу інструментів відносяться системи «Jason», «Eclipse AMP».

Яскравим представником такого класу систем є «Repast Symphony» – модуль до системи розробки (IDE) «Eclipse». Така реалізація дозволяє використовувати існуючі можливості середовища, надаючи при цьому

методи розробки «розумних» агентних систем, котрі дозволяють сконцентруватися саме на написанні «розумних» агентів, використовувати при цьому наочні та цікаві засоби.

Головною перевагою є безпосередня робота з «розумними» агентами і системами їх взаємодії. Існують прості методи створення «розумних» агентів, завдання їх параметрів з використанням блок-схем. Все це робить роботу з системою досить зручною та зрозумілою. В середовищі взаємодії з користувачем присутні ряд недоліків, зокрема, неможливість відкату при здійсненні деяких важливих операцій, складності при збереженні та відновленні, часткової недопрацьованості системи (часом з'являються помилки, що перешкоджають роботі системи, відстежити які може тільки розробник), громіздкість і часто зайвий обсяг виконуваних операцій.

Крім того, незважаючи на початкову легкість створення програм в такому середовищі, сам процес створення є строго регламентованим.

Все це робить необхідним періодичне використання низькорівневих засобів розробки та суворе дотримання задекларованих схем побудови «розумних» мультиагентних систем. Такий стан зменшує переваги поєднання стандартних і потужних засобів розробки зі спеціалізованими методами.

1.5 Висновок до першого розділу

У цьому розділі було:

- Подано перлік задач в проектах класу «розумне місто» котрі можуть бути вирішені з використанням «розумних» мультиагентних систем.
- Наведено визначення «розумного» програмного агента, а так само проведено аналіз існуючих «розумних» програмних агентів.
- Розглянуто інструментарій для створення «розумних» програмних агентів, який включає мови і програмні засоби реалізації «розумних» агентів, мови комунікації «розумних» агентів, мови опису поведінки «розумних»

агентів і законів середовища, мови представлення та управління знаннями, мови формалізації і специфікації «розумних» агентів і MAC.

– Розглянуто мови програмування «розумних» програмних агентів, в результаті чого можна зробити висновок, що при програмуванні «розумних» агентів найчастіше застосовуються мови об'єктно-орієнтованого програмування («C++», «Java»), рідше використовуються символічні мови і мови логічного програмування (LISP, Oz).

2 ЗАСОБИ СПЕЦИФІКАЦІЙ ТА РОЗРОБЛЕННЯ «РОЗУМНИХ» МІСЬКИХ ПРОГРАМНИХ АГЕНТІВ

2.1 Засоби специфікації моделей «розумних» програмних агентів

Типова модель «розумного» програмного агента отримала аббревіатуру RAMM та є подальшим розвитком моделі програмного агента Shoham, у якому всі дії виконуються тільки як результат певних зобов'язань. В межах моделі «розумного» програмного агента RAMM ця ідеологія розширена до рівня формування множини загальних правил поведінки, які визначають модель функціонування агента в кожному конкретному випадку. Разом з тим правила поведінки «розумного» агента функціонують на зафіксованій множині його «реакцій» відповідно до поточного стану навколишнього середовища. [25]

Узагальнена схема процесу проектування та практичної реалізації «розумних» агентно-орієнтованих застосунків побудованих на основі середовища AgentBuilder ToolKit подана на рисунку 2.1. [26]

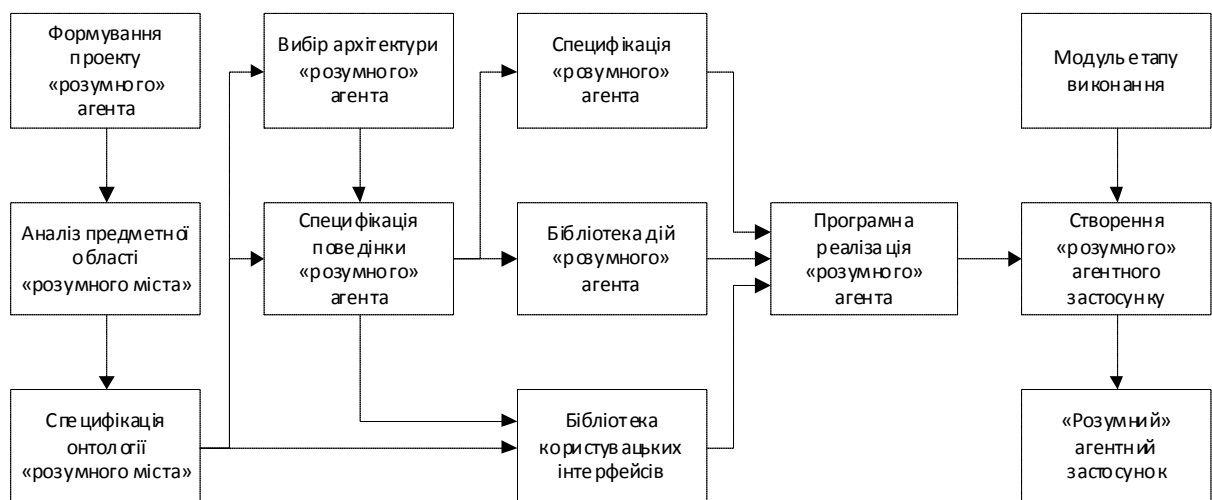


Рисунок 2.1 – Інформаційно-технологічна схема протікання процесу розроблення «розумних» агентно-орієнтованих застосунків на базі середовища AgentBuilder ToolKit

Зазначений інструментарій має засоби для організації множини описів створюваної предметної області MAC, засоби для формування специфікації архітектури і моделей поведінки «розумних» агентів, засоби налагодження «розумних» агентних застосунків та інструментальні засоби для організації спостереження за поведінкою створених «розумних» агентів.

Для розроблення специфікації поведінки «розумних» агентів в середовищі AgentBuilder використовується спеціалізована об'єктно-орієнтована мова специфікацій RADL. При цьому правила поведінки можуть розглядатись у вигляді конструкції «WHEN-IF-THEN». В яких «WHEN»-частина нотації правила призначена для опису нових подій, що виникають в середовищі-оточенні «розумного» агента та містять нові інформаційні повідомлення, отримані від інших «розумних» програмних агентів.

«IF»-частина використовується для порівняння поточної ментальної моделі «розумного» агента з умовами застосування логічних правил. Перелік зразків в «IF»-частинах використовується для виконання «намірів», «зобов'язань» і «можливостей», перелік яких визначено поточною ментальною моделлю.

«THEN»-частина нотації правила визначає перелік дій, як реакцію у відповідь на зміну подій та станів поточної ментальної моделі та навколишнього оточення. Зазначені правила можуть включати оновлення поточної ментальної моделі «розумного» агента, комунікативні та внутрішні реакції.

На рисунку 2.2 подано модель «життєвого циклу» «розумного» агента побудованого в середовищі AgentBuilder. [27]

Для використаної в середовищі мові RADL активно використовуються інформаційно-логічні прототипи, доступні набори досить розвинених засобів для роботи з наборами змінних та представлення множини дій, що включає функціональні можливості для формування та використання перформативів засобами мови KQML. [28]



Рисунок 2.2 – Модель «життєвого циклу» «розумного» агента сформована в середовищі AgentBuilder

Інформаційні структури даних, для яких відбувається застосування засобів по суті є фреймами а дана мова підтримується засобами поданими на інструментальному рівні з використанням спеціальних мовно-орієнтованих засобів та редакторів інтегрованих в системі.

Специфікація поведінки «розумних» агентів та сформованих для них «розумних» ментальних моделей поміщається в окремий спеціальний файл збережений у форматі «adf», який використовується сумісно з класами та методами із попередньо сформованої бібліотеки дій «розумних» агентів та інтерфейсів. Для інтерпретації зазначеного файлу використовується програмна компонента RRTAE, котра є частиною виконання процесів у середовищі AgentBuilder. [29]

В процесі оцінювання підходів до формування специфікацій моделей поведінки «розумних» агентів, котрі використовуються в середовищі AgentBuilder, слід відзначити загальну складність системи подання та маніпулювання накопиченими знаннями призначеної для опису моделей типу RAMM. При цьому в згаданій «розумній» моделі відсутні засоби для організації виводу, призначені для збільшення мовних функціональних можливостей. В зазначеній моделі відсутні засоби для явного фіксування

стану «розумного» агента, котрі відмінні від бінарних перемикачів для значень змінних. При цьому не визначено методику врахування в специфікації доступних «розумних» моделей поведінки функціональних можливостей реалізації різних «ліній поведінки», що є характерною особливістю дійсно «розумних» агентів. Не достатньо обґрунтовано використання особливих режимів інтерпретації різних реалізацій поведінки «розумних» агентів. При цьому слід відзначити, що інструментарій середовища AgentBuilder містить надзвичайно потужні та сучасні засоби проектування та практичної реалізації MAC і може бути ефективно використаний в процесі реалізації інформаційно-технологічних проектів «розумних міст».

2.2 Інструментальні засоби для розроблення та побудови «розумних» програмних агентів

Інструментальні засоби для розроблення «розумних» програмних агентів використовуються в процесах формування середовище, оптимізованого для створення «розумних» застосунків із специфічною та адаптивною архітектурою. [30]

Основна відмінність «розумних» інструментальних середовищ від інших прикладних засобів побудови «розумних» програмних агентів полягає в забезпеченні повного циклу розроблення «розумних» програмних агентів за рахунок навколишнього середовища «розумного міста». При цьому включаються етапи аналізу предметної області «розумного міста», включаючи етапи проектування, розроблення, верифікації, розгортання та їх супроводу. Виділимо найбільш популярні та поширені середовища розробки «розумних» агентів:

1. JADE.
2. JACK.

3. BAE.

4. Bee-gent.

Розглянемо детальніше особливості перелічених інструментальних середовищ для розроблення та практичної реалізації «розумних» програмних агентів.

2.2.1 Середовище реалізації «розумних» програмних агентів JADE

Програмне середовище для реалізації «розумних» програмних агентів JADE отримало обширне використання в процесах комплексного розроблення багатоагентних «розумних» систем. Середовище реалізовано з використанням мови Java з можливістю підтримання FIPA-стандарту створення «розумних» агентів. Метою створення JADE-середовища було спрощення процесів розроблення на основі процесів стандартизації способів взаємодії «розумних» агентів у глобальному середовищі реалізації «розумних» системних сервісів. [31]

Для досягнення поставленої мети JADE-середовище пропонує програмістам та розробникам «розумних» агентних систем наступні функціональні можливості:

1. Агентно-орієнтовану платформу FIPA-compliant, побудовану на основі FIPA. Котра з метою підвищення ефективності їх управління включає обов'язкові процеси типізації «розумних» системних агентів, «розумну» агентну платформу (AMS), доступні канали комунікації (ACC) та інтегровані служби каталогів (DF). Перелічені інструменти для реалізації «розумних» агентів будуть автоматично активованими при першому запуску платформи.

2. Розподілену агентно-орієнтовану платформу (DAP), яка може одночасно використовувати декілька хостів. При цьому на кожному інтегрованому вузлі запускається тільки одна (JVM), а «розумні» агенти виконуються у вигляді окремих Java-потоків. В залежності від місцезнаходження конкретного «розумного» агента, відбувається надсилання

сервісного повідомлення. [32] Для доставки сервісних повідомлень використовується інтегрований з платформою транспортний механізм.

3. MDS – ряд «розумних» агентів базованих на FIPA DF може бути об'єднано в федерацію з подальшою реалізацією мультидоменного «розумного» агентно-орієнтованого середовища.

4. Багатопотокове середовище виконання «розумних» агентів з дворівневим плануванням потоку команд та подій. Кожен із зазначених «розумних» JADE-агентів має власний та неповторний потік управління. При цьому він може працювати в багатопотоковому режимі. JVM проводить планування потоку завдань, що передаються до виконання одному або декільком «розумним» агентам.

5. Об'єктно-орієнтоване програмне середовище. Більшість концепцій, котрі є характерними для FIPA-специфікації, представляються у формі Java-класів, що використовуються для формування інтерфейсу користувача.

6. Бібліотека інтегрованих протоколів взаємодії дозволяє використання стандартних інтерактивних протоколів «fipa-request» та «fipa-contract-net». З метою створення «розумного» агента, який може діяти відповідно до зазначених протоколів, розробникам прикладних програмних застосунків потрібно виконувати імплементування специфічних доменних операцій. В той самий час вся незалежна від прикладного програмного рівня протокольна логіка буде реалізовуватися з використанням системи JADE.

7. Адміністративний GUI. Для спрощення виконання операцій управління платформою використовується графічний інтерфейс, що відображає поточний стан активних «розумних» агентів разом з їх контейнерами. Використовуючи GUI, системні адміністратори платформи можуть виконувати операції створенні, знищенні, переривання та відновлення функціонування «розумних» агентів, створювати при цьому ієрархії доменів та мультиагентні федерації на основі DF-фасилітаторів.

Програмне середовище JADE базується на інформаційних технологіях «Java Serialization» і «Java Reflection API», «Java RMI» з використанням «Java CORBA IDL». В цьому середовищі процедура розробки MAC спрощується завдяки інтегрованому використанню FIPA-специфікацій та обширного переліку інструментів для підтримування фаз налагодження та розгортання інформаційної системи інтегрованої в проекти класу «розумне місто». Зазначена інформаційно-технологічна платформа може бути встановлена на комп'ютерах з обширним переліком операційних систем та сконфігурована з використанням віддаленого GUI-інтерфейсу. [33] Процес конфігурації розглянутої інформаційно-технологічної платформи досить гнучкий та може бути змінений під час виконання програмних застосунків за рахунок переміщення «розумних» агентів з однієї фізичної обчислювальної машини на іншу. При цьому залишається чи не єдина вимога щодо обов'язкового встановлення «Java Run Time» версії не нижче 1.2.

Кожен окремий запущений екземпляр системного середовища JADE є окремим контейнером та може містити множину з декількох «розумних» агентів. Об'єднана група активних контейнерів, при цьому, утворює платформу в якій завжди повинен бути активним головний контейнер. Всі інші контейнери, при їх створенні, повинні бути зареєстровані на його основі. В такому випадку перший контейнер, запущений на системній платформі буде основним, а всі інші – дочірніми, або звичайними контейнерами і обов'язково повинні отримувати вказівки щодо місцезнаходження їх основного контейнера, на якому відбувається їх реєстрація. У випадку, якщо в мережі відбувається запуск ще одного основного контейнера, то він буде реалізацією іншої платформи, на буде доступна реєстрація нових звичайних контейнерів. На рисунок 2.3 подано описані вище концепції системної платформи і контейнера та продемонстровано сценарій з двома JADE-платформами, що складаються відповідно з одного та трьох контейнерів. [34]

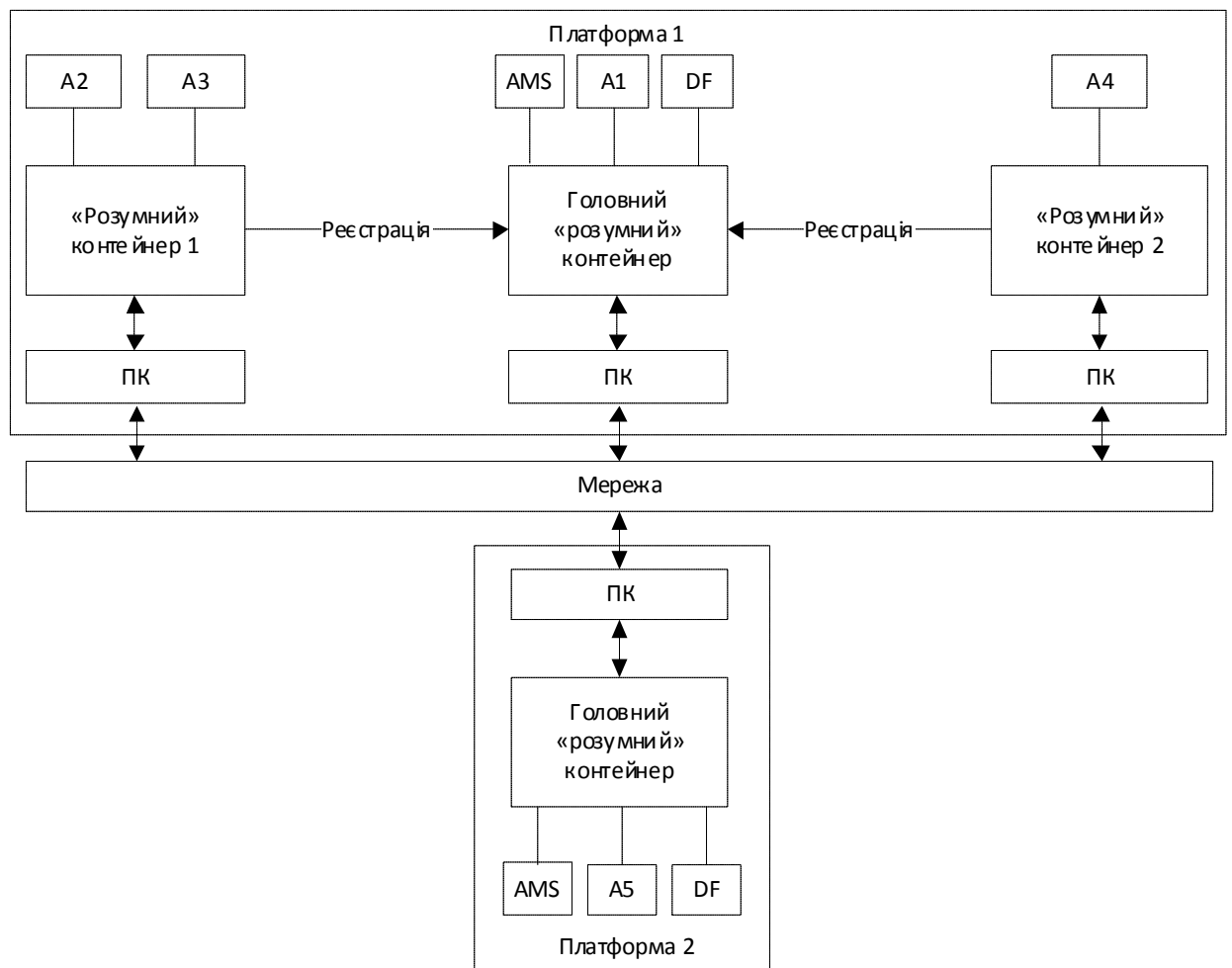


Рисунок 2.3 – Приклад реалізації JADE-середовища «розумних» агентів

Середовище JADE базується на інформаційних технологіях з використанням «Java RMI» та «Java C».

JADE «розумні» агенти повинні мати свої унікальні імена та знати імена сусідів. Що дозволить забезпечувати безпосереднє спілкування, котре не залежить від місця їх фактичної локалізації, в межах одного виокремленого контейнера (зокрема, наприклад, агенти A2 і A3), в різних контейнерах всередині однієї окремої платформи (наприклад, A1 і A2) або в різних платформах (наприклад, A4 і A5).

Основний «розумний» контейнер відрізняється від решти тим, що містить розгорнуту систему управління «розумними» агентами і маршрутизатор, які автоматично інтегруються та запускаються при його першому запуску. Система управління «розумними» агентами AMS виконує

ключові функції управління елементами платформи, зокрема виконує операції створення та видалення «розумних» агентів запитуваних через AMS і забезпечує функціонування служби їх іменування. Маршрутизатор служби каталогів DF забезпечує функціонування сервісу «Жовтих сторінок» та допомагає «розумному» агенту знайти інших «розумних» агентів, з метою отримання від них запитуваних послуг, необхідних для досягнення сформованих в процесі функціонування цілей.

Для здійснення міжагентної комунікації архітектура «розумного» середовища надає гнучкий і ефективний процес обміну повідомленнями, в якому JADE створює чергу і управляє потоком приватних для кожного «розумного» агента ACL-повідомлень. «Розумні» агенти можуть надсилати до черги запити за допомогою комбінування декількох режимів роботи: блокування, голосування, перерва в роботі та порівняння з еталоном (у випадку використання методів пошуку). 35]

В останніх версіях системи використовується «Java RMI», «event-notification» та «IIOP». Проте, можна легко додати будь-які інші протоколи. Передбачена можливість інтеграції мережових протоколів SMTP, HTTP та WAP. В системі інтегрована та доступна переважна більшість комунікаційних протоколів визначених міжнародною спільнотою розробників «розумних» агентних середовищ. Їх використання проілюстровано на конкретних практичних прикладах після визначення моделі поведінки системи та ідентифікації основних її станів. [36]

В інформаційній системі реалізовані онтології високого рівня, котрі використовуються для управління «розумними» агентами та онтології низького рівня, які реалізуються та реєструються агентами в подальшому використовуються системою. З метою розширення функціональних можливостей середовища JADE передбачено можливість його інтеграції з «JESS» і Java-оболонкою «CLIPS».

2.2.2 Середовище реалізації «розумних» програмних агентів JASK

Інтегроване програмне середовище «JASK TM Intelligent Agents» (JASK) являє собою агентно-орієнтоване середовище розроблення, побудоване з використанням мови програмування Java. JASK є Java-надбудовою реалізованою у вигляді синтаксичного розширення Java-конструкціями з метою програмної реалізації функціональних можливостей пов'язаних з «розумними» агентами. [37]

Мова програмування «розумних» агентів JASK пропонує наступні функціональні можливості:

1. Визначає нові основні агентно-орієнтовані класи, інтерфейси і методи.
2. Розширює Java-синтаксис для підтримки нових агентно-орієнтованих класів, визначень і операторів.
3. Надає семантичні розширення (особливо при виконанні) для підтримки моделі виконання, необхідної для «розумної» агентно-орієнтованої програмної системи.

Всі розширення реалізовані у вигляді плагінів, що додає мові максимальної гнучкості в «розумному» агентно-орієнтованому програмуванні. На рівні класів введено 5 ключових структурних конструкцій:

1. Агент, який в JASK моделює «розумну» сутність.
2. Можливості для об'єднання в одне ціле всіх функціональних компонентів, зокрема, подій, планів, множину упереджень і ін.
3. Події для моделювання ситуацій і повідомлень, на які «розумні» агенти повинні мати функціональні можливості реагувати та відповідати.
4. Призначений для моделювання процедурного опису план керування наперед визначеними для «розумного» агента подіями.
5. Множину упереджень та переконань, для моделювання знань «розумного» агента у вигляді сформованих переконань, які дотримуються семантичного опису закритого або відкритого зовнішнього або внутрішнього

світу. Зазначена конструкція формує переконання «розумного» агента у вигляді реляційних кортежів першого порядку та забезпечує їх повну логічну несуперечливість.

Слід відзначити, що бажана поведінка «розумного» агента буде інкапсульована в модульних одиницях, визначених поданими вище класами котрі містять всі необхідні для незалежного виконання «розумні» структури і методи, які можуть бути використані програмістами. Для формування та встановлення відносин між переліченими вище класами існує набір декларацій. В лістингу 2.1 подано фрагмент програмного коду для реалізації конструкції процедурного плану, написаного на мові програмування «розумних» агентів JASK. [38]

Лістинг 2.1 – Фрагмент програмного коду з елементами мови JASK

```
«plan MovementOneResponse extends for Plan {
#handles event OneRobotOneMoveEvent moveresponse;
#uses smart agent implementing OneRobotOneInterface robot;
static boolean relevantings (OneRobotOneMoveOneEvent ev)
{ ... }
context() { ... }
#reasoning this method
body(){ ... }}.»
```

У цьому прикладі відбувається визначенні план дій для «розумного» програмного агента, котрий успадковує основні виконувані функції від класу JASKPlan. При цьому, за допомогою декількох декларацій для планів мови JASK вказується, яким чином буде використовуватися поточний план. Кожна декларації починається символом «#» з метою виділення від елементів синтаксису Java. Декларація «#handles event» визначає мету або подію, для якої формується відповідність зазначеного плану. Декларація «#uses smart agent implementing» ідентифікує «розумних» агентів, які можуть використовувати зазначений план. Поданий в прикладі план можуть виконувати тільки агенти, які реалізують зазначений інтерфейс «OneRobotOneInterface». У фігурних дужках міститься звичайний Java-код.

Для опису використовуваних «розумним» агентом міркувань, переконань і поведінки при виконанні поточного плану мова JASK, крім декларацій надає свої оператори для опису методів та міркувань, які виділяються символом «@» поміщеним на початку нотації. [39]

Для підтримки процесів виконання в «розумному» агентно-орієнтованому програмному середовищі JASK надає перелік наступних додаткових мовних розширень, що забезпечують наступну семантику:

1. Вбудована в ядро багатопоточність, котра виведена з-під контролю програмістів.

2. Функціонування «розумних» агентів здійснюється таким чином, що агенти опрацьовують множину планів та отримують доступ до описів переконань. «Розумні» агенти виконують наперед сформовані плани в задачах керування опрацюванням подій, у випадку їх виникнення, порівнюючи сформовані для кожного з них переконання у випадку необхідності. Зазначені плани можуть ініціювати виконання дочірніх задач, які також можуть ініціювати дочірні задачі.

3. Введена нова «розумна» структура даних подана у формі логічного елемента «logical member», значення якого безпосередньо залежить від результатів запиту сформованого до множини переконань агента.

4. Можливість виконання запитів сформованих до множини переконань агента з використанням логічних елементів та операцій їхнього об'єднання. У випадку успішного виконання запиту логічний елемент буде містити бажане значення.

Компонент середовища розробки «JASK Development Environment» дає можливість створення структурних діаграм. На основі яких середовище згенерує каркас програмного коду та стежить за узгодженістю відображення між кодом та діаграмами.

Агенти, створені засобами JASK, мають архітектуру, притаманну «розумним» агентам. Таким чином, можна моделювати «розумну» поведінку,

агента відповідно до теоретичної моделі його BDI-архітектури сформованої на основі його переконань, бажань і намірів. [40]

Відповідно до BDI-архітектури «розумні» агенти в середовищі JACK – це автономні «розумні» програмні компоненти, які можуть демонструвати «розумну» поведінку на основі проактивності (цілеспрямованість) і реактивності (направляється подіями) на входні подразники та сигнали. Кожен такий «розумний» програмний агент має:

- переконання (це його набір даних про навколишній світ);
- бажання (набір подій на які він буде реагувати і набір цілей, досягнення яких він буде прагнути);
- наміри (набір планів, які описують процеси управління виникаючими цілями та планами).

Якщо «розумного» програмного агента розглядати як аналог фізичної особистості, то набір планів описує послідовність дій при виникненні певної події або формуванні бажання спрямованого на досягнення певного результату. На перший погляд, поведінка «розумного» агента може видатися схожою на функціонування експертних систем, разом з усіма властивими їм функціональними та структурними обмеженнями. Однак, принципова відмінність «розумних» агентно-орієнтованих систем в тому, що «розумні» агенти можна запрограмувати для виконання планів подібно до «розумної» особистості. З використанням «розумних» програмних агентів можна реалізувати властивості, що асоціюються з «розумною» поведінкою:

- стійку цілеспрямованість – «розумні» агенти зосереджені на сформованих цілях, а не на вибраних методах їх досягнення;
- контекстна залежність в режимі реального часу – агенти стежитимуть за варіантами застосовними в кожен певний момент часу і прийматимуть рішення щодо подальших дій, на основі сформованого переліку наявних умов;

– фіксування правильності підходу в режимі реального часу – «розумний» агент гарантуватиме слідування обраному курсу відповідно до істинності обраних певних умов.

– одночасність – «розумна» агентна система володіє властивістю багатопотоковості. «Розумний» агент здатний формувати та визначити пріоритети на вимогу багатозадачності у випадку виникнення нових цілей та подій.

JACK-застосунок є вихідним кодом, який реалізує характерні для «розумного» агентно-орієнтованого підходу сутності, зокрема, агенти та здібності, події та плани, переконання та запити. Також реалізується Java-клас з функцією «main()», яка є точкою входу для віртуальної Java-машини та будь-які інші необхідні Java-файли. Імена створених для цих сутностей файлів повинні мати відповідні їм імена. Розширення імен файлів визначають тип JACK-сутності. Компілятор «розумних» JACK-агентів конвертує вихідні файли сформовані на мові «розумних» JACK-агентів в програмний Java-код, який потім компілюється у виконуваний на цільовій системі код віртуальної Java-машини. [41]

2.2.3 Середовище AgentBuilder

Середовище AgentBuilder надає програмістам та розробникам інструментальні засоби для реалізовані в середовищі виконання «розумних» агентів та застосунків [42]. Методика створення «розумного» агента засобами середовища AgentBuilder подана на рисунку 2.4.

Засоби розроблення і середовище виконання написані з використанням мови програмування Java. Це дозволяє роботу створених «розумних» програмних агентів на всіх інформаційно-технологічних платформах, де встановлено середовище Java. Відповідно «розумний» агент, створений за допомогою інструментарію AgentBuilder, може виконуватися на будь-якій платформі з віртуальною машиною Java (версії 1.1 і вище).

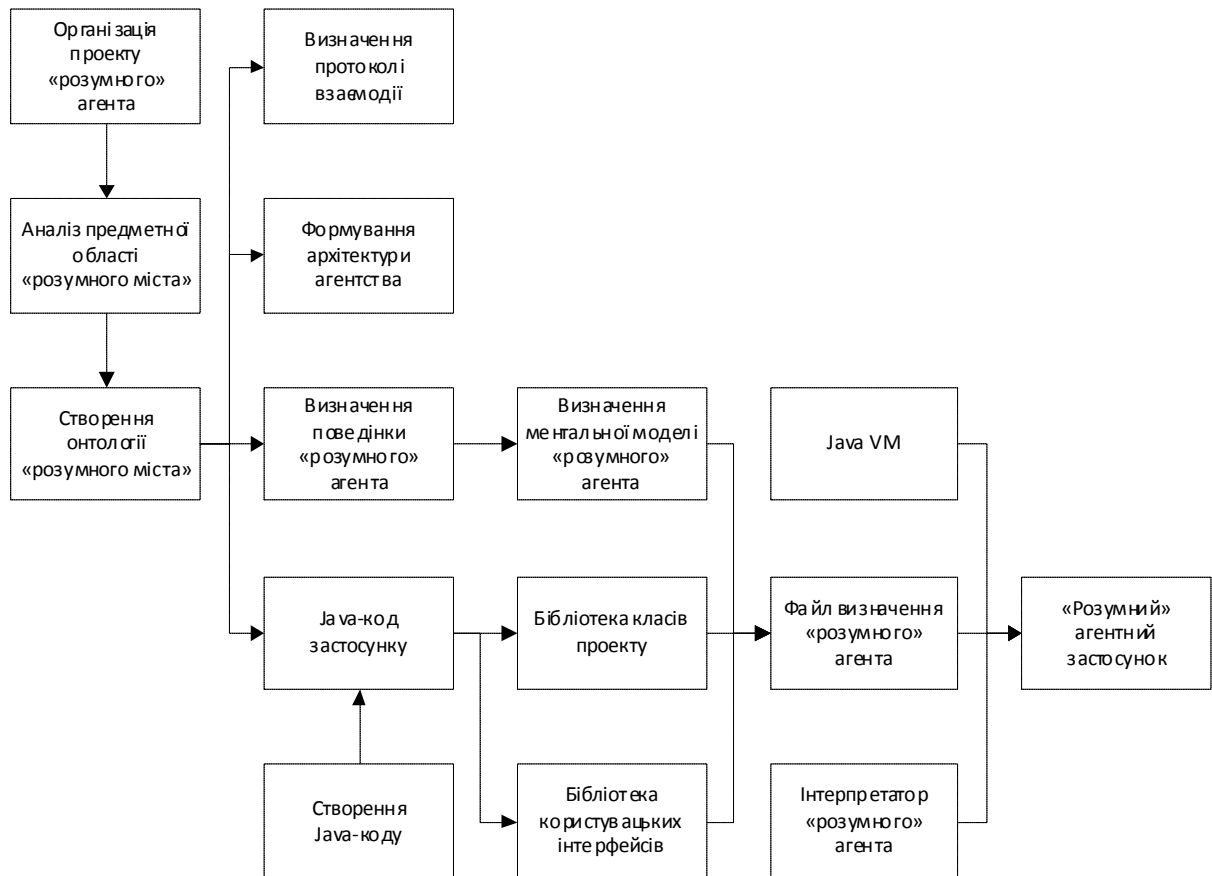


Рисунок 2.4 – Методика створення «розумного» агента з використанням інструментарію середовища AgentBuilder

Засоби розробки представляють собою зручний графічний інтерфейс для аналізу предметної області «розумного міста» для якої розробляється проект MAC і створюються специфікації бажаної поведінки «розумних» агентів, що розробляються за допомогою графічних редакторів. В інструментальному середовищі передбачені наступні етапи побудови багатоагентних «розумних» застосунків [43]:

1. Визначення переліку «розумних» агентів та їх складу.
2. Створення «розумних» агентів передбачає побудову онтології для виконання делегованих повноважень і ментальної моделі, котра містить переконання та здібності, зобов'язання та правила поведінки.
3. Створення протоколів для специфікації взаємодії групи «розумних» агентів.

4. Генерація спеціального файлу опису «розумного» агента на мові RADL, який містить ментальну модель та модель поведінки агента.

Середовище виконання «розумного» мультиагентного застосунка складається з «розумної» агентної програми і процесора виконання «розумного» агента. Процесор використовує ефективні процедури логічного виведення шляхом співставлення правил поведінки «розумного» агента з моделлю його переконань, описаними поточною ментальною моделлю та сформованою вхідними повідомленнями. На основі проведених логічних виведень процесор виконує дії пов'язані з повноваженнями та поведінкою агента. «Розумна» агентна програма являє собою визначення агента у вигляді файлу записаного в нотації мови RADL разом з укомплектованою бібліотекою класів «розумного» проекту. «Розумна» агентна програма спільно з процесором утворюють модель виконуваного «розумного» агента. При запуску середовища виконання, ініціалізується процесор «розумного» агента, який використовує RADL-модель і онтологію «розумного» агента, подану у вигляді бібліотеки класів проекту (PAL). Для цього необхідні визначення «розумного» агента (файл RADL, який забезпечує «розумного» агента здатністю виконувати міркування відповідно до початкової ментальної моделі та бібліотеки класів «розумного» проекту (допоміжні класи проекту PACs видобуті з бібліотеки класів «розумного» проекту) – ці об'єкти використовуються для відображення предметної області «розумного міста».

2.2.4 Середовище Bee-gent

У середовищі Bee-gent розробка «розумних» агентно-орієнтованих застосунків виконується з використанням методології специфікації поведінки «розумних» агентів розподіленої системи з використанням MAC – бібліотеки, реалізованої засобами мови програмування Java. На основі запропонованих системою Bee-gent графічних засобів, можлива чітка структуризація поведінки кожного «розумного» агента поданого у вигляді

графа станів та визначення протоколів взаємодії «розумних» агентів. Графи станів «розумних» агентів будуються на основі ролей, визначених у формі регулярних виразів на етапі «розумного» агентно-орієнтованого аналізу проведеного наприклад, за методологією Gaia [44]. Приклад фрагменту графа поведінки «розумного» агента «Користувач інновацій» інформаційної системи інтегрованої в проект «розумного міста» поданий на рисунку 2.5.

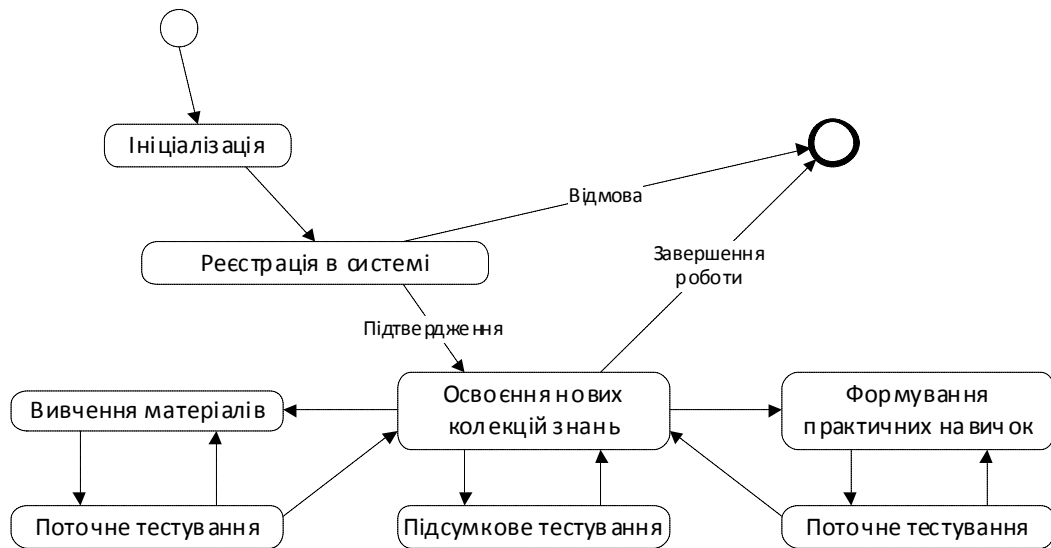
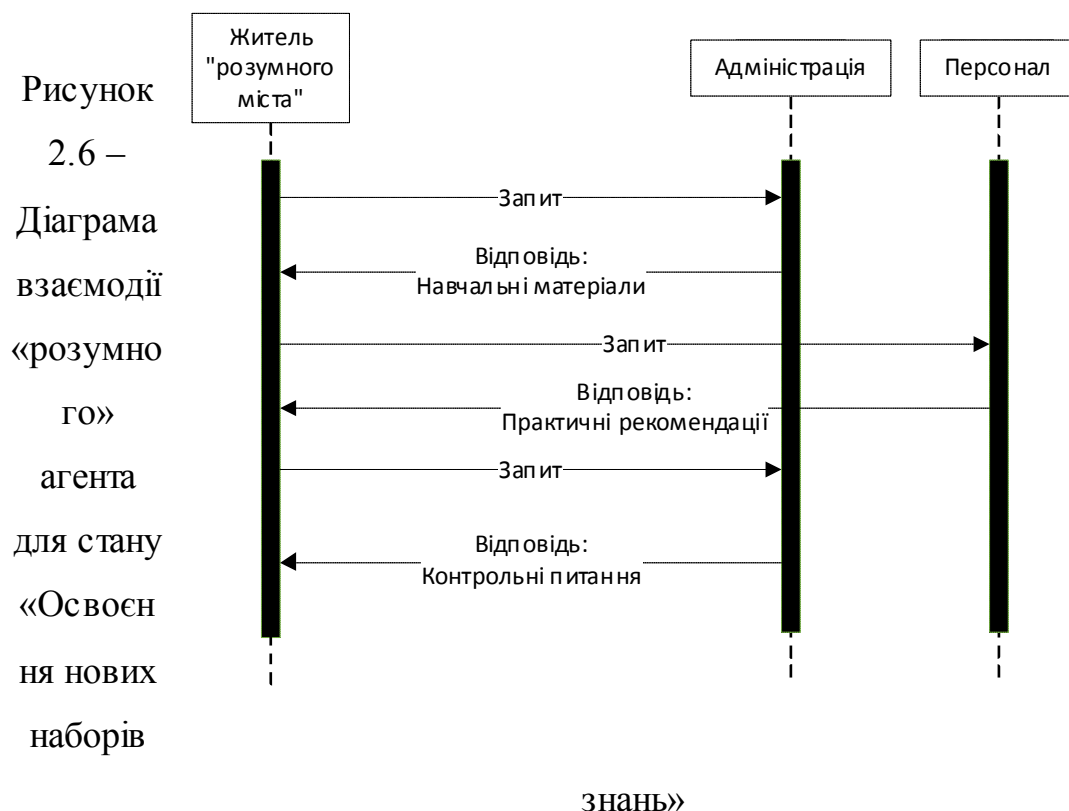


Рисунок 2.5 – Граф станів поведінки «розумного» агента
«Користувач інновацій»

Поданий на рисунку граф станів реєструє всі імена станів, в яких «розумний» агент може знаходитись. Наступним етапом розроблення буде визначення класів для кожного стану. Кожне стан поданий графом є екземпляром класу «AwrIPState» з агентної бібліотеки фірми «Toshiba», реалізованої засобами мови Java. У конструкторі класу визначаються перед та пост умови призначені до виконання «розумним» агентом в поточному стані для виконання дії визначеної класом стану і відбувається визначення переходу в наступний стан. Потім відбувається специфікація дій, які повинні бути виконані в кожному стані, включаючи власні процеси «розумного» агента та взаємодії з іншими «розумними» агентами. Для початкового і кінцевого станів «розумного» агента створюються класи

«SMART_AGENT_INIT» і «SMART_AGENT_END». Якщо «розумний» агент взаємодіє з іншими «розумними» агентами, то при специфікації окремих станів інформаційна система Bee-gent передбачає визначення протоколу міжагентної взаємодії. Протокол повинен відображати всі можливі лінії поведінки «розумного» агента в даному стані. У кожному з можливих станів діяльність «розумного» агента спрямована на виконання протоколів взаємодії з метою реалізації планованої лінії його поведінки. Діяльність кожного «розумного» агента в MAC визначається, наприклад, моделлю послуг, розробленою на етапі агентно-орієнтованого аналізу за методологією Gaia.

Кожна лінія поведінки документується з використанням діаграми взаємодії «розумних» агентів із зазначенням допустимого вмісту повідомлень та протоколу їх черговості. На рисунку 2.6 подано приклад діаграми взаємодії для стану «Освоєння нових наборів знань». Формат повідомлень визначається транскрипцією записаною з використанням мови «XML/ACL», яка є подальшим розвитком мови комунікації «KQML». [45]



Таким чином, на основі розроблених логічних моделей, система «Bee-gent» автоматично генерує на мові Java-каркас програмного коду багатоагентної «розумної» системи, який буде доповнюватись програмним кодом необхідним для всіх етапів життєвого циклу «розумних» агентів. В системі «Bee-gent», на відміну від «AgentBuilder», при описі моделей поведінки «розумних» агентів не застосовуються правила, що визначають реакцію «розумного» агента на зовнішні події зміну його внутрішнього стану. [46]

2.3 Порівняльний аналіз інструментальних середовищ

Порівняльний аналіз функціональних можливостей розглянутих інструментальних середовищ для розроблення «розумних» програмних агентів подано в таблиці 2.1.

Таблиця 2.1 – Порівняльний аналіз можливостей інструментальних середовищ для розробки «розумних» програмних агентів

Можливості інструментальних середовищ	JADE	JACK	Agent Builder	Bee-gent
1	2	3	4	5
Засоби побудови груп «розумних» агентів	+	-	+	-
Засоби управління проектами	-	+	+	-
Графічне середовище для визначення специфікацій агентів	-	+	+	-
Механізм контролю цілісності	-	+	+	-
Засоби побудови онтології	-	-	+	-
Бібліотека для розробки MAC	+	+	+	+
Механізм міркувань агента про свої здібності і здібності інших агентів	-	+	+	-
Формальна мова комунікації	+	+	+	+

Продовження таблиці 2.1

1	2	3	4	5
Засоби відлагодження взаємодії агентів	+	+	+	-
Механізм пошуку агентів з заданими здібностями	+	-	+	-
Оцінка можливостей інструментальних середовищ в %	42%	71%	92%	21%

На рисунку 2.7 в графічному вигляді подані результати проведеного аналізу.

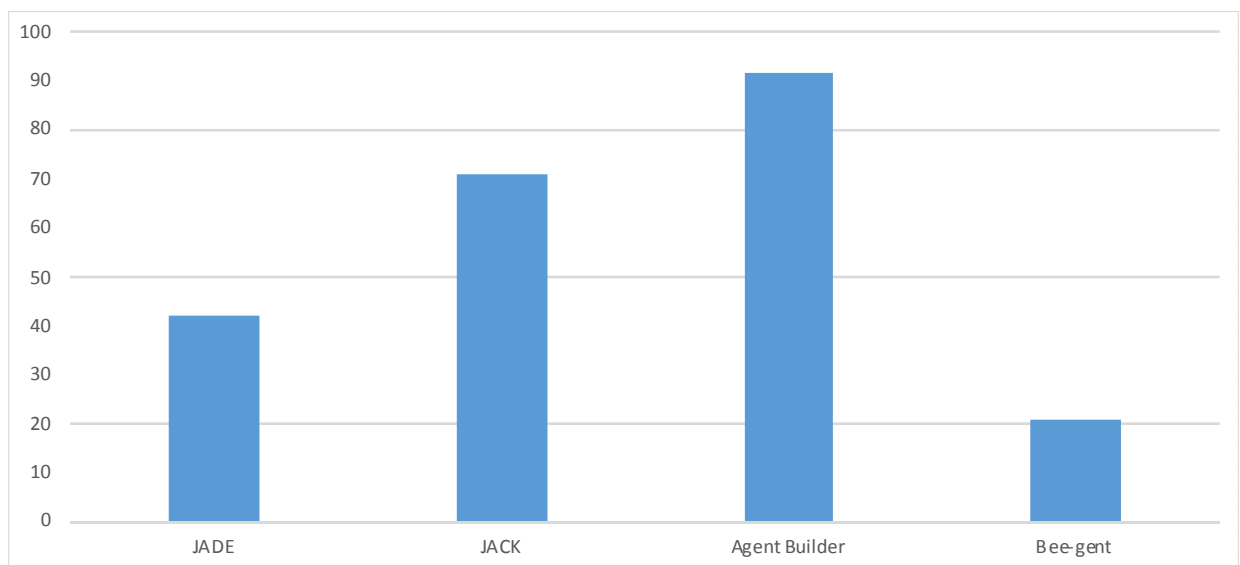


Рисунок 2.7 – Графічне подання проведеної оцінки можливостей інструментальних середовищ у %

На підставі порівняння характеристик розглянутих інструментальних середовищ для розроблення «розумних» агентів можна зробити висновок, що найбільш потужними і технологічно-гнучкими засобами реалізації «розумних» агентів є інструментарій AgentBuilder та середовище JACK.

Необхідно звернути увагу на те, що для платформи JADE існує додаткове BDI розширення реалізоване у вигляді середовища Jadex. Зазначене середовище надає гібридну реактивно-деліберативну архітектуру,

в якій «розумний» агент розглядається у вигляді «чорного ящика», який приймає та надсилає повідомлення. Ґрунтуючись на результатах опрацювання повідомлень, внутрішніх станів та зовнішніх подій, деліберативний механізм приймає рішення щодо переходу до нового плану дій або продовження виконання старого. Чинний поточний план дій може передбачати надсилання повідомлень іншим агентам, внесення змін до бази переконань, формування нових цілей та ініціалізацію виклику процедур опрацювання внутрішніх подій. Інформаційна система використовує бібліотеку планів дозволяючи їх опрацювання у вигляді Java-класів.

Однією з головних переваг від розроблення «розумних» агентів на базі інформаційно-технологічної платформи Jadex є відсутність потреби щодо вивчення нових мов програмування. Замість цього «розумні» агенти кодуються засобами об'єктно-орієнтованого програмування в інтегрованому середовищі розроблення (IDEs), типу «Eclipse і IntelliJ IDEA». [47]

Ще одним важливим аспектом є незалежність розробленого програмного забезпечення. Оскільки інформаційно-технологічна платформа Jadex є незалежною разом з модулями, то вона може використовуватися в абсолютно різних сценаріях на верхньому рівні платформи. Агентно-орієнтовані «розумні» програми додають нові властивості для реалізації автономних структурних елементів в процесах прийняття рішень по відношенні до пасивних об'єктів. У цьому відношенні «розумні» агенти надають активні компоненти для реалізації індивідуальних можливостей взаємодії з компонентами.

Інформаційно-технологічна платформа Jadex розроблена у вигляді самостійного механізму прийняття рішень, адаптованого для роботи з будь-якими системами «розумного міста», які виконують взаємодію з «розумним» агентом щодо управління його поведінкою за допомогою повідомлень.

Програмна-реалізація «розумного» агента може вільно мігрувати між хостами, виконуючи операції, як на стороні сервера, так і на стороні

користувача та зберігаючи при цьому незалежність від місця виконання поставлених завдань.

Проведений аналіз найбільш відомих інструментальних систем розроблення «розумних» агентів дозволив вибрати ефективне та доступне середовище Jadex.

2.4 Висновок до другого розділу

Широта досліджень в області мультиагентних «розумних» систем, складність інформаційно-технологічної архітектури та різноманіття програмних компонент «розумних» агентів, різні методології подання та розроблення MAC спричинили розроблення великої кількості програмних засобів для їх розроблення.

У цьому розділі були докладно проаналізовані та розглянуті найбільш популярні середовища для розроблення «розумних» агентів та проведено порівняльний аналіз їх можливостей. Зокрема були досліджені середовища для розроблення «розумних» програмних агентів JADE та JACK, BAE та «Bee-gent». В результаті проведеного аналізу можна зробити висновок, що на сьогоднішній день найбільш розвинутим, ефективним та доступним середовищем є інформаційно-технологічна платформа JADE в сукупності з Jadex.

3 АРХІТЕКТУРА ТА МОДЕЛІ ПРОГРАМНИХ АГЕНТІВ У «РОЗУМНОМУ МІСТІ»

3.1 Архітектура мультиагентних застосунків «розумного міста»

У загальному випадку середовище «розумного міста» в якому відбувається функціонування «розумних» агентів, має певні повністю або частково відомі правила та закономірності поведінки. Стан середовища «розумного міста» залежить від наявності у «розумного» агента інформації, його дискретності стану та детермінованості дій, динамічності або статичності та синхронності або асинхронності зміни стану і т.п.

На відміну від традиційних інформаційних систем, в яких пошук рішення відбувається за допомогою централізованих, послідовних і детермінованих алгоритмів, в «розумних» мультиагентних системах рішення досягається як результат розподіленої взаємодії множини «розумних» агентів поданих у вигляді автономних програмних сутностей, цілеспрямованих на пошук не зовсім оптимального, але «найкращого» з можливих логічних рішень у визначений момент часу. Якщо знайдений «розумним» агентом кращий варіант вже заброньовано іншим «розумним» агентом то «розумні» агенти можуть бути здатні вирішити поточний конфлікт інтересів шляхом діалогу, спрямованого на досягнення компромісу, щодо тимчасового відображення, і, як правило сформулювати нестійкий баланс інтересів. [48]

Зазначений принцип радикально відрізняє отримані в такий спосіб рішення і дозволяє застосовувати мультиагентні «розумні» інструменти, продукти та системи для вирішення найскладніших завдань.

В цілому, мультиагентні «розумні» системи або агентно-орієнтоване програмування є наступним етапом у розвитку об'єктно-орієнтованого програмування (ООП) котре інтегрує досягнення останніх десятиліть у сфері

штучного інтелекту, паралельних обчислень і телекомунікацій. Розрізняють три базових типи архітектури «розумних» МАС:

1. Архітектура на основі принципів і методів роботи зі знаннями.
2. Архітектура на основі поведінкових моделей.
3. Гібридні побудови, що поєднують в різних співвідношеннях особливості двох перших архітектур.

З класичної точки зору архітектура на основі знань це «розумна» архітектура, яка містить символічну модель світу подану в явній формі, і в якій прийняття рішень щодо дій, які повинні бути виконані «розумним» агентом, здійснюється на основі логічних або псевдо-логічних міркувань. Такий «розумний» агент може розглядатися як спеціальний випадок системи, побудованої на знаннях.

Спочатку ідея «розумного» агента, заснованого на знаннях, будувалася на чисто логічній основі і видавалася дуже перспективною. Однак пізніше було виявлено, що формування на основі такого підходу з множиною предикатів першого порядку нерозв'язне. Більш того, такі ментальні властивості «розумного» агента, як переконання, бажання, наміри, зобов'язання по відношенню до інших «розумних» агентів і т.д. не можуть бути подані в термінах предикатів першого порядку. Були розроблені спеціальні варіанти розширень модальних логік. Такі архітектури були названі BDI-архітектурами. [49]

У системах, побудованих на основі поведінкових моделей, ключовим поняттям є «ситуація», в якій опинився «розумний» агент, і «спонукальний мотив» до вироблення адекватної в цій ситуації дії або поведінки («реакції») «розумного» агента. Поведінкові моделі найчастіше будуються на правилах вироблення адекватних реакцій. Або, якщо різноманітність ситуацій не є катастрофічною – застосовуються алгоритми, що дозволяють створити кінцевий автомат з обмеженими можливостями. «Розумні» агенти МАС даного типу називаються реактивними, а їхня реакція на той чи інший

спонукальний мотив завжди розглядається як результат співвіднесення стану зовнішнього середовища і внутрішнього стану «розумного» агента. Найбільш перспективні гібридні МАС, що поєднують «кращі» якості поведінкових і продукційних моделей. Гібридні МАС часто використовують принцип спеціалізації своїх «розумних» агентів. Примітно, що БЗ «розумних» агентів таких систем має знання трьох рівнів – про предметну область «розумного міста», про можливі взаємодії з іншими «розумними» агентами (якщо своїх недостатньо) і спеціальні знання, що забезпечують оптимальне управління «розумною» мультиагентною системою.

3.2 Модель ресурсів «розумного» програмного агента

«Розумний» програмний агент в мультиагентній системі «розумного міста» має свій опис у вигляді BDI-моделі, яка містить його знання, плани і цілі, які «розумний» агент виконує для досягнення своїх завдань в середовищі «розумного міста». Розглянемо опис BDI моделі котра відображає опис цілей «розумного» агента, знань і його планів.

Однією з найцікавіших і широко розповсюджуються архітектур «розумного» агента є архітектура «Цілі-Факти-Плани» (BDI). Ця архітектура складається з суджень та знань, фактів і цілей поданих у вигляді згенерованих уявних відношень.

Обрана платформа Jadex об'єднує модель «розумних» агентів Jade, вводячи цілі, факти і плани для маніпуляції всередині «розумного» агента. У Jadex, «розумні» агенти мають знання, які можуть бути яким-небудь видом збереженим в БЗ Java-об'єктів. [50] Цілі відображають конкретні напрями спонукування переходу до певного стану.

Плани являють собою певні дії агента в середовищі функціонування «розумного міста» і задаються розробником системи, формуючи бібліотеку дії планів виконання «розумним» агентом. Залежно від ситуації виконання

гіперскладної системи «розумного міста», в якій функціонує «розумний» агент, вибирається відповідний план дії на виконання «розумним» агентом. Вибір планів здійснюється автоматично платформою Jadex.

Цілі в Jadex є об'єктами з певними наборами атрибутів. Цільовий стан, спрямований на досягання мети, може бути явно представлено логічним виразом, який досягнення мети.

На відміну від традиційних BDI-систем, які розуміють цілі як специфічний вид подій, в Jadex елементи мети є центральною концепцією. Цілі – це конкретні, миттєві бажання «розумного» агента. Будь-яка мета, яку переслідує «розумний» агент, буде більш-менш безпосередньо здійснена відповідним дією, поки не буде припущення, що поставлена мета досягнута, недосяжна або більше не бажана. [51]

Jadex підтримує 4 види цілей:

1. Мета виконання – стверджує, що має бути виконано, але не обов'язково приведе до певного результату.

2. Мета досягнення – описує абстрактний цільовий стан, який необхідно досягнути, без визначення алгоритму роботи. Таким чином, «розумний» агент може випробувати різні альтернативні варіанти, для досягнення поставленої мети.

3. Мета запиту – представляє необхідність в інформації. Якщо інформація на поточний момент не доступна, вибираються і виконуються плани, які допомагають її зібрати та отримати.

4. Мета підтримки – специфікує стан, який повинен зберігатися, і може викликатися, після того як буде виконаний.

Цілі становлять мотиваційну установку «розумного» агента, яка є рушійною силою його дій. Тому можна зробити висновок, що уявлення і обробка цілей – це одна з головних функцій платформи Jadex.

Знання представляють обізнаність «розумного» агента про навколишнє середовище «розумного міста» та про себе безпосередньо. У платформі Jadex

знання можуть бути подані будь-якими об'єктами java. Вони зберігаються в БЗ і можуть бути згадані в виразах, доступні для зміни в залежності від планів, що використовують БЗ застосунків. Знання описуються в ADF-файлі і звертаються до зміненого плану, визначаючи єдину оцінку плану відповідно до необхідних для нього результатів. [52]

Розглянемо схему роботи «розумного» програмного агента ресурсів (див. рисунок 3.1).

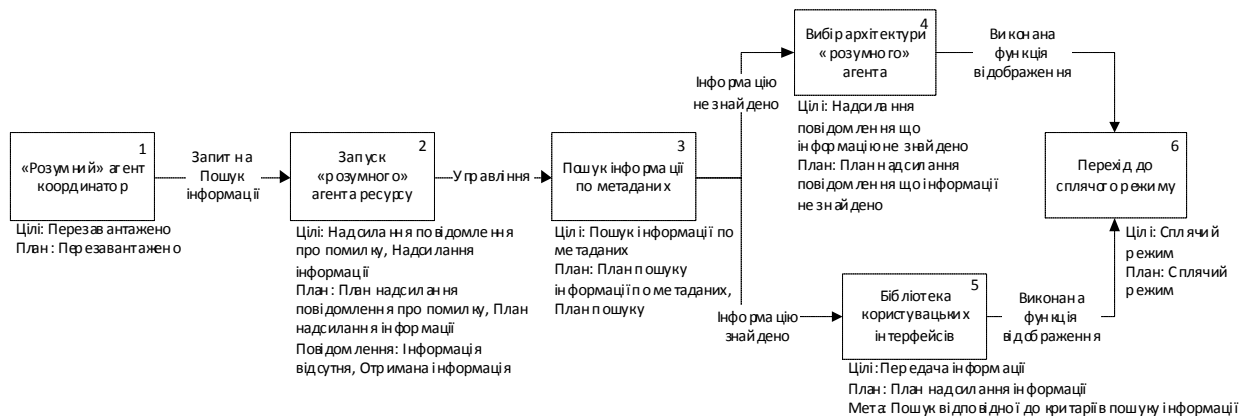


Рисунок 3.1 – Схема роботи «розумного» програмного агента

Робота «розумного» агента починається з того, що «розумний» агент координатор (1 блок моделі) посилає «розумному» агенту ресурсу запит на пошук інформації. Далі запускається «розумний» агент ресурсів (2 блок моделі), після подачі запиту на пошук інформації, яку шукає користувач за допомогою інших «розумних» агентів. Після цього управління передається в наступній блок моделі «розумного міста» для пошуку необхідної інформації по метаданих. «Розумний» агент координатор керує взаємодією між усіма «розумними» агентами.

В даній схемі можливі два варіанти подій, при яких шукана інформація або знайдена, або ні. Якщо інформацію не знайдено, «розумному» агенту координатору буде відправлено повідомлення про те, що даний ресурс відсутній (блок 4). Якщо ж інформація знайдена (блок 5), тоді «розумному» агенту координатору перенаправляється матеріал, який був знайдений. При

виконанні блоку 4 або 5 функція «розумного» агента вважається виконаною. Після чого «розумний» агент переходить в сплячий режим (блок 6), до того часу поки не з'явиться нова необхідність в його роботі.

На прикладі схематичною розгляної моделі (див. рис 3.1) наочно розглянута схема роботи «розумного» програмного агента ресурсів «розумного міста». Крім цього на даній схемі описані цілі, факти і плани, які «розумний» агент буде досягати в процесі своєї роботи.

3.3 Опис розробленого програмного модуля «розумного» агента

Jadex – це «розумний» агент, орієнтований на власний механізм прийняття рішень, котрий використовує взаємодію з використанням XML і Java файлів для підтримки прийнятих рішень.

Платформа Jadex призначена для створення «розумних» агентів, які взаємодіють з гіперскладною системою «розумного міста», до якої підключений «розумний» агент і змінюють події складових систем «розумного міста» в залежності від задекларованих агентом планів та цілей.

Щоб створити «розумного» агента необхідно визначити в гіперскладній системі «розумного міста» властивості агента.

Дії «розумного» агента описуються за допомогою уявлень, цілей та планів. Помістивши їх, в бібліотеку подану у вигляді XML-файлу, котрий називається ADF-файлом визначення «розумного» агента [53]. У ADF-файлі «розумний» агент визначаються як об'єкт, описаний у вигляді набору java-класів. Запуск «розумного» агента визначається в активних планах, які викликаються на вимогу гіперскладної системи «розумного міста».

Функціонування в «розумного» агента в гіперскладній системі «розумного міста» відбувається за рахунок отримання від системи та «розумного» агента відправка повідомлень. Система подає на вхід «розумному» агенту повідомлення, після чого «розумний» агент заносить

його в так званий «модуль подій». Передаючи його в блок обробки повідомлень, блок планів та блок аналізу. Подія аналізується в контексті приналежності «розумного» агента. Якщо повідомлення не належить «розумному» агенту, то воно передається назад в систему без змін. Блок планів при опрацюванні події, порівнює їх з власною базою планів «розумного» агента і при позитивному порівнянні з планами починає обробку повідомлення в базі знань, звідки надходить для порівняння з цілями «розумного» агента для передачі в модуль подій. Пройшовши весь цикл порівняння і перетворення даних, в установлені завдання, надходить в систему «розумного міста» для подальшого функціонування.

Інтерпретатор складається зі списку інформації компонента основних планів вирішення завдання, зокрема це так звані заплановані мета та дії, орієнтовані на виконання «розумним» агентом. Режим роботи приблизно виглядає так: «розумний» агент вибирає мета дію з ADF-файлу і виконує його, коли виконуються попередні умови дії. В іншому випадку дія пропускається. Виконання дії може призвести до виконання подальші дії, які додані до списку ADF-файлу після налаштування «розумного» агента відповідно до концепції основних планів вирішення глобального завдання та плану. В даний момент часу, концепції дій головним чином розрізняють пов'язані і непов'язані дії, відповідно, пов'язані дії додаються у формі дочірніх (породжених) вузлів до поточного вузла плану.

Розглянемо базовий алгоритм побудови «розумного» агента побудованого з використанням платформи Jadex. Розробляючи «розумний» застосунок з використанням Jadex, програміст зобов'язаний створювати два типи файлів: XML-файли позначаючи «розумного» агента і Java-файли для виконання планів. Файл визначення «розумного» агента використовується як специфікація класів виконання плану, котра базується на оголошеннях дій цього плану та основних фактах взаємозв'язку між ними. Файли в форматах XML та Java визначають функціональні можливості «розумного» агента.

У головному XML-файлі «розумного» агента визначаються всі функціональні можливості, якими він буде наділений. Зокрема це: імпорти – «imports», можливості – «capabilities», переконання – «beliefs», цілі – «goals», плани – «plans», події – «events», вирази – «expressions», властивості – «properties», елементи ініціалізації – «initial states». Коли ADF і java завантажені, об'єкти створені для XML в ADF, дозволяють «розумній» системі взаємодіяти, використовуючи Jadex-платформу для прийняття рішень щодо певних дій і взаємодії всередині «розумного» програмного модуля. «Розумні» агенти мають уявлення щодо їх оточення і завдань, до яких вони прагнуть. Для досягнення мети «розумний» агент виконує плани, які є процедурами, кодованими у формі Java-нотацій.

Найактуальніша і широко поширена архітектура «розумного» агента – це зведена архітектура «Цілі-Факти-Плани» (BDI). Дана архітектура складається з сутностей знань, фактів і цілей поданих у формі уявних відношень, які генеруються людськими діями. Ця модель була прийнята та перетворена в формальну теорію і архітектуру виконання для «розумних» програмних агентів, побудованих на основі фактів, цілей, і планів. Jadex об'єднує цю модель для «розумних» агентів Jade, вводячи цілі, факти і плани для маніпуляції всередині «розумного» агента. У Jadex, «розумні» агенти мають знання, які можуть бути яким-небудь видом Java-об'єктів, і запам'ятали в базу знань. Цілі зображують конкретні прагнення спрямовані на досягнення певного стану.

3.4 Висновок до третього розділу

У розділі подано архітектуру «розумних» мультиагентних застосунків «розумного міста». Наведено опис програмного модуля для побудови моделі програмного агента з використанням технології Jadex, яка дозволяє моделювати «розумні» BDI-агенти з наборами фактів, цілей і планів.

Крім цього розділі розглянуто «розумний» програмний агент у вигляді BDI-моделі і запропонована схематична модель роботи «розумного» програмного агента ресурсу, для якого описані цілі, факти і плани, які агент буде досягати в процесі своєї роботи. Подано опис розробленого та використаного програмного забезпечення.

4 СПЕЦІАЛЬНА ЧАСТИНА

4.1 Реалізація моделі міського «розумного» програмного агента

У даній магістерській роботі для побудови програмного модуля використовується технологія Jadex, яка дозволяє моделювання «розумних» BDI-агентів з наборами фактів та цілей, планів та інших атрибутів. У даній магістерській роботі подані «розумні» програмні агенти, які застосовуються для забезпечення підтримки персоналізованого навчання міських жителів в умовах «розумного міста».

Розроблюваний модуль «розумного» агента «BazaZnan» містить модулі по формуванню термінів «BazaZnanDictionary.xml» до нього додається визначення типу документа «BazaZnanDictionary.dtd», котрий визначає перелік блоків XML-документа, його структуру з переліком допустимих елементів (атрибутів). Також містить модуль логічної відповіді користувачу «BazaZnanAgent.xml».

Вся база знань представлена в XML-файлі, а опис його тегів контролюється в DTD-файлі. В даному XML-файлі під назвою «BazaZnanDictionary.xml» описано перелік термінів, які будуть використовуватися «розумним» агентом при пошуку матеріалу, якого навчають студенту.

«<? Xml version = "1.0"?>» – тег позначає версію документа. «<!DOCTYPE bazaznan SYSTEM "GlossaryDictionary.dtd">» – даний рядок показує, що ми визначаємо структуру документа XML в DTD-форматі. На початку XML-файл описується заголовковим тегом, що визначає його призначення «<bazaznan> ... </ bazaznan>». Описуємо назву за допомогою атрибута «title = "...» і дату створення «orderDate = "...? ...? ..."». В ньому визначаються терміни «<terms> ... </ terms>». Властивості терміна описуємо, використовуючи, теги: «<term> ... </ term>» – тег, яким оголошуємо термін.

«value = "term ..."» – значення терміна при пошуку в переліку матеріалу. Далі оголошуємо ім'я за допомогою тега «<name> ... </ name>». «<Image ref = "...>» – даний тег описує рисунки до того терміну, який йому відповідає. «<Count> ... </ count>» – відповідає за кількість визначень терміну. А в свою чергу визначення позначаються як «<definition> ... </ definition>», і вже в ньому описуємо «count = "...» яке це визначення за рахунком. Також існують вкладені гіперпосилання – «<giperlink> ... </ giperlink>», які відповідають за те, що б при перегляді терміну, коли людина переходить на вкладену посилання агент міг відреагувати на цей запит.

Опис «розумного» агента БЗ відбувається також через xml-формат. У ньому описуються цілі, поставлені перед «розумним» агентом. Також описуються плани, факти, події за допомогою яких «розумний» агент досягає цілей. Початковий опис «розумного» агента БЗ можна побачити на нижче викладених рисунку 4.1.

BazaZnaniAgent.xml

```
<!--<H3>BazaZnani agent</H3>-->  
<agent xmlns="http://jadex.sourceforge.net/jadex"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://jadex.sourceforge.net/jadex  
http://jadex.sourceforge.net/jadex-0.94.xsd"  
name="BazaZnani"  
package="BazaZnani">
```

Рисунок 4.1 – Загальний опис «розумного» агента бази знань

Перелік бібліотек для імпорту для «розумного» агента подано на рисунку 4.2.

```

<imports>
  <import>jadex.planlib.*</import>
  <import>jadex.runtime.*</import>
  <import>jadex.util.*</import>
  <import>jadex.adapter.fipa.SFipa</import>
  <import>java.util.logging.*</import>
  <import>jadex.adapter.fipa.*</import>
</imports>|

```

Рисунок 4.2 – Імпортовані бібліотеки

Перелік цілей «розумного» агента подано на рисунку 4.3.

```

<goals>
  <!-- Query goal represents a necessity in information, executes the search of
  terms. -->
  <querygoal name="query_term" exclude="never">
    <parameter name="result" class="term" direction="out">
      <value evaluationmode="dynamic">
        select one Sterm from Sbeliefbase.terms
        where !Sterm.isFull()
        order by Sbeliefbase.my_location.getDistance(Sterm.getLocation())
      </value>
    </parameter>
  </querygoal>
  <!-- Main goal of the student is to give additional information. -->
  <maintaingoal name="term_definition">
    <maintaincondition>
      Sbeliefbase.query = true
    </maintaincondition>
  </maintaingoal>
</goals>

```

Рисунок 4.3 – Цілі роботи «розумного» агента

Приклад використаних «розумним» агентом знань подано на рисунку 4.4. Вони описують множину виконуваних фактів.

```

<beliefs>
<!-- Restart agent -->
<belief name="Restart" class="BazaZnanii">
<fact>BazaZnaniiPlanRestart();</fact>
</belief>
<!-- The presence of request for additional information.
Search of terms from the list of materials. -->
<belief name="QuerySearch" class="BazaZnanii">
<fact>BazaZnaniiPlanSearch.getBazaZnaniiDictionary();</fact>
</belief>
<!-- Opening of the found searching information. -->
<belief name="ContentOpen" class="BazaZnanii">
<fact>BazaZnaniiPlanOpenDictionary();</fact>
</belief>
<!-- Send message to agent coordinator. -->
<belief name="FileNotFound" class="BazaZnanii">
<fact>BazaZnaniiPlanSendMessageTermNotFound();</fact>
</belief>
<!-- Send message to agent chat. -->
<belief name="NoTerm" class="BazaZnanii">
<fact>BazaZnaniiPlanSendMessageNoTerm();</fact>
</belief>
<!-- Send message to student. -->
<belief name="NoReference" class="BazaZnanii">
<fact>BazaZnaniiPlanSendMessageNoReference();</fact>
</belief>
<!-- Sleep mode agent-->
<belief name="SleepMode" class="BazaZnanii">
<fact>BazaZnaniiPlanSleep();</fact>
</belief> </beliefs>

```

Рисунок 4.4 – Використовувані «розумним» агентом знання

Приклад використаних «розумним» агентом планів подано на рисунку 4.5.

```

<plans>
  <!-- Plan restart agent-->
  <plan name="restart">
    <body>new BazaZnanüPlanRestart()</body>
  </plan>
  <!-- Plan presence of references in the text-->
  <plan name="reference">
    <body>new BazaZnanüPlanReference()</body>
  </plan>
  <!-- Plan search of necessary term-->
  <plan name="search">
    <body>new BazaZnanüPlanSearch()</body>
    <trigger><message event ref="request_reference"/></trigger>
  </plan>
  <!-- Plan reflection of content-->
  <plan name="content_open">
    <body>new BazaZnanüPlanOpenDictionary()</body>
  </plan>
  <!-- External plan. Description of interaction with an agent by a coordinator.-->
  <plan name="coordinator">
    <body>new BazaZnanüPlanSendMessageTermNotFound()</body>
    <trigger><message event ref="request_inform"/></trigger>
  </plan>
  <!-- External plan. Description of interaction with an agent by a chat.-->
  <plan name="chat">
    <body>new BazaZnanüPlanSendMessageNoTerm()</body>
    <trigger><message event ref="request_failure"/></trigger>
  </plan>
  <!-- Plan show message to student-->
  <plan name="student">
    <body>new BazaZnanüPlanSendMessageNoReference()</body>
    <trigger><message event ref="request_reference"/></trigger>
  </plan>
  <!-- Plan sleep mode agent-->
  <plan name="sleep"> <body>new BazaZnanüPlanSleep()</body> </plan> </plans>

```

Рисунок 4.5 – Виконувані плани «розумного» агента

Приклад використаних «розумним» агентом подій подано на рисунку 4.6.

```

<events>
  <messageevent name="request_inform" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
      <value>SFipa.INFORM</value>
    </parameter>
    <parameter name="content-start" class="String" direction="fixed">
      <value>"Error! Reference of the term is not found."</value>
    </parameter>
  </messageevent>
  <messageevent name="request_failure" direction="receive" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
      <value>SFipa.FAILURE</value>
    </parameter>
    <parameter name="content-start" class="String" direction="fixed">
      <value>"It is necessary to lay out material. Add dictionary" </value>
    </parameter>
  </messageevent>
  <messageevent name="request_reference" direction="send" type="fipa">
    <parameter name="performative" class="String" direction="fixed">
      <value>SFipa.REQUEST</value>
    </parameter>
    <parameter name="content-start" class="String" direction="fixed">
      <value>"Reference is not found. Make an effort other time." </value>
    </parameter>
  </messageevent>
</events>

```

Рисунок 4.6 – Події, що виводяться повідомленнями для передачі іншим «розумним» агентам або кінцевому користувачеві

Приклад опису початкового стану «розумного» агента і завершення його ініціалізації подано на рисунку 4.7.

```

<initialstates>
  <initialstate name="default">
    <plans>
      <initialplan ref="SeachTerm" />
    </plans>
  </initialstate>
</initialstates>
</agent>

```

Рисунок 4.7 – Опис початкового стану «розумного» агента і завершення його ініціалізації

Розроблений в магістерській роботі «розумний» програмний агент поданий у вигляді BDI-моделі з множиною створених фактів, цілей, планів, повідомлень та розробленою схемою взаємодії модулів «розумного» агента і сформованими зв'язками з іншими «розумними» агентами в середовищі проекту «розумного міста». З використанням платформи «Jadex» для початку «розумний» агент описується у вигляді HTML-нотації відповідно до заданих планів, цілей і фактів.

Зокрема «розумний» агент має характеристики:

- наявність запиту та пошуку додаткової інформації;
- відкриття знайденої інформації котра відповідає критеріям пошуку;
- перегляд необхідної інформації.

Вхідна інформація:

- запит на пошук необхідного матеріалу.

Вихідна інформація:

- знайдена за посиланням інформація.

У розробленого «розумного» агента виділяється дві мети. Це цілі виду запит («query») і головна мета («main goal»). Мета запиту зображує необхідність в інформації та виконує пошук умов, являє необхідність в інформації. Головна мета – це надати жителям та гостям «розумного міста» додаткову інформацію. Ця мета описує стан, який повинно бути досягнуто.

У знаннях формуються описи фактів виявлених в ході роботи агента. У даному випадку описується сім знань, в яких визначаються факти. При запуску агента викликається факт «Restart agent». Наступним фактом є запит на пошук інформації «QuerySearch», котрий показує присутність запиту на додаткову інформацію та виконує пошук термінів. Відкриття шуканої інформації з словника виконується за допомогою факту «ContentOpen». «TermNotFound» – говорить про те, що буде відправлено повідомлення про помилку «розумному» агенту координатору. А «розумному» агенту чату відправляється повідомлення за допомогою факту «NoTerm» про те, що необхідно викладення необхідного матеріалу, якого немає в наявності на даний момент часу. Відсилається повідомлення користувачу «розумного міста» про помилкове посилання. «SleepMode» – показує перехід «розумного» агента в сплячий режим.

У розробленого «розумного» агента виділяються плани перезавантаження – «restart», довідка – «reference», пошук – «search», відкрити матеріал – «content_open», координатор – «coordinator», зовнішній план – «chat», користувач – «user», сон – «sleep»:

- «Restart» – виконує запуск агента при виклику його агентом координатора.

- «Reference» – присутність посилань по тексту.

- «Search» – пошук необхідного терміну.

- «Content_open» – відображення вмісту словника.

- «Coordinator» – опис взаємодії з агентом координатором. Відсилає повідомлення про помилку, коли термін не знайдено. Є зовнішнім планом.

- «Chat» – зовнішній план. Опис взаємодії з агентом чату. Відсилає повідомлення про те, що матеріалу немає в наявності.
- «User» – вивід повідомлення користувачу. Відсилає повідомлення про те, що посилання неправильне і матеріалу немає в наявності.
- «Sleep» – агент переходить в сплячий режим.

Таким чином, була розроблена модель «розумного» агента, яка визначає його дії в загальній мультиагентній системі «розумного міста». Також були визначені цілі для досягнення виконання «розумним» агентом, факти та плани для виконання цих цілей.

4.2 Висновок

В цьому розділі відображено реалізацію моделі міського «розумного» програмного агента. Зокрема подано опис «розумного» агента, нотацію його цілей та знань, планів та подій. Наведено приклад опис його початкового стану та характеристики.

5 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Метою дипломної роботи освітнього рівня «Магістр» є дослідження методів та засоби реалізації програмних агентів для проектів класу «розумне місто». Головною метою розділу є встановлення економічної доцільності проведення даної розробки.

Щоб виконати оцінку економічної ефективності необхідно розрахувати трудомісткість реалізації дослідження, витрати на оплату праці найманим працівникам, витрати апаратного і програмного забезпечення, амортизаційні відрахування, витрати енергоресурсів та інші витрати які є основними пунктами виконання обчислень, а також показники економічної ефективності проведеного дослідження.

5.1 Розрахунок норм часу на виконання науково-дослідної роботи

Реалізація дослідження методів та засобів реалізації програмних агентів для проектів класу «розумне місто» складається з низки послідовних та взаємопов'язаних етапів.

Кожен із етапів реалізації дослідження характеризується метою та змістом, оцінкою часу виконання, кількістю та спеціалізацією виконавців, а також приблизною оцінкою вартості.

Дослідження методів та засобів реалізації програмних агентів для проектів класу «розумне місто» складається із підготовчого етапу, етапу технічної пропозиції, створення технічного завдання, проектування системи, практичної реалізації, тестування, верифікації та заключного етапу.

Норми часу на виконання науково-дослідницької роботи розраховуватимуться на основі середнього часу виконання стадії в годинах, що наведені в таблиці 5.1 разом із інформацією про виконавців і сумарною кількості затраченого часу.

Таблиця 5.1 – Операції науково-дослідного процесу та час їх виконання

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1.	Підготовча стадія	Проектний менеджер	2
		Інженер-програміст	
2.	Технічна пропозиція	Проектний менеджер	80
		Інженер-програміст	
3.	Створення технічного завдання	Проектний менеджер	84
		Інженер-програміст	
4.	Проектування системи	Проектний менеджер	82
5.	Практична реалізація	Інженер-програміст	50
6.	Тестування системи	Тестувальник	50
7.	Верифікація системи	Проектний менеджер	30
		Інженер-програміст	
		Тестувальник	
8.	Створення документації	Інженер-програміст	20
9.	Заключна стадія	Проектний менеджер	20
Разом			418

В підсумку на дослідження методів та засобів реалізації програмних агентів для проектів класу «розумне місто» необхідно 418 людино-годин, залучення трьох спеціалістів та виконання дев'яти різноманітних стадій реалізації проекту.

5.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Визначення витрат на оплату праці та відрахувань на соціальні заходи прямо залежить від кількості витраченого працівниками часу на роботу,

ставки в годину чи місяць, кількість відрахувань на соціальні заходи встановлених в законному порядку на час розрахунку.

В результаті розрахунку потрібно визначити основну та додаткову заробітну плату, витрати на соціальні заходи та на основі цих даних визначити сумарні витрати на оплату праці.

Основна заробітна плата нараховується за виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов’язані з виплатами за фактично відпрацьований час.

При розрахунку заробітної плати кількість робочих днів у місяці слід в середньому приймати – 24,5 дні/міс., або ж 196 год./міс. (тривалість робочого дня – 8 год.).

Наймані працівники для розробки інформаційної системи управління доступом з використанням інформаційних технологій розпізнавання образів працюють згідно контракту, який в якому вказано їхню погодинну ставку. Тобто розрахунок заробітної плати працівників відбуватиметься на базі тарифної ставки та кількості відпрацьованих годин.

У штаті найманих працівників для розробки інформаційної системи залучено проектного менеджера, інженера-програміста і тестувальника.

Тарифні ставки учасників процесу розробки інформаційної системи управління доступом з використанням інформаційних технологій розпізнавання образів:

- Проектний менеджер – 150 грн./год.
- Інженер-програміст – 130 грн./год.
- Тестувальник – 100 грн./год.

Основна заробітна плата розраховується за формулою:

$$Z_{осн.} = T_c \cdot K_z, \quad (5.1)$$

де T_c – тарифна ставка, грн.;

K_c – кількість відпрацьованих годин.

Оскільки всі види робіт в виконує три спеціалісти, то основна заробітна плата буде розраховуватись за даною формулою 5.1.

$$Z_{осн.} = 150 \cdot 195 + 130 \cdot 163 + 100 \cdot 60 = 56440,00 \text{ грн.}$$

Додаткова заробітна плата становить 10–15 % від суми основної заробітної плати й визначається за формулою 5.2. Коефіцієнт додаткових виплат працівникам становить 0,1.

$$Z_{дод.} = Z_{осн.} \cdot K_{дод.}, \quad (5.2)$$

де $K_{дод.}$ – коефіцієнт додаткових виплат працівникам.

$$Z_{дод.} = 56440,00 \cdot 0,1 = 5644,00 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{о.п.}$ – фонд заробітної плати) визначаються за формулою 5.3:

$$B_{о.п.} = Z_{осн.} + Z_{дод.} \quad (5.3)$$

$$B_{о.п.} = 56440,00 + 5644,00 = 62084,00 \text{ грн.}$$

З цієї суми утримуються обов'язкові відрахування на заробітну плату:

- Єдиний соціальний внесок (ЄСВ), що становить – 22%.
- Військовий збір (ВЗ), що становить – 1,5%.

Сума відрахувань становить 23,5 % від фонду оплати праці та визначається за формулою 5.4:

$$B_{с.з.} = \Phi_{ОП} 0,235, \quad (5.4)$$

де $\Phi_{ОП}$ – фонд оплати праці, грн.

$$B_{с.з.} = 62084,00 \cdot 0,235 = 14589,74 \text{ грн.}$$

Усі витрати обчислюються детально наведені в таблиці 5.2 та обчислюються за формулою 5.5.

$$B_{з.п.} = \Phi_{ЗП} + \Phi_{ОП}, \quad (5.5)$$

$$B_{з.п.} = 56440,00 + 5644,00 + 14589,74 = 76673,74 \text{ грн.}$$

Таблиця 5.2 – Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівни- ків	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахув. на ФОП, грн.	Всього витрати на оплату праці, грн. $6=3+4+5$
		Тарифна ставка, грн.	К-сть відпра- цьов. год.	Фактично нарах. з/пл., грн.			
А	Б	1	2	3	4	5	6
1	Проектний менеджер	150	195	29250,00	2925,00		
2	Інженер- програміст	130	163	21190,00	2119,00		
3	Тестуваль- ник	100	60	6000,00	600,00		
Разом		380	418	56440,00	5644,00	14589,74	76673,74

Опираючись на розрахунки витрат на оплату та зведену таблицю результатів 5.2 видно, що всього витрати на плату праці становлять 76673,74 грн.

5.3 Розрахунок матеріальних витрат

Матеріальні витрати є невід’ємною частиною дослідження методів та засобів реалізації програмних агентів для проектів класу «розумне місто» та визначаються як добуток кількості витрачених матеріалів та їх ціни за формулою 5.6:

$$M_{Bi} = q_i \cdot p_i, \quad (5.6)$$

де: q_i – кількість витраченого матеріалу i -го виду;

p_i – ціна матеріалу i -го виду.

Звідси, загальні матеріальні витрати можна визначити за формулою 5.7:

$$Z_{мв.} = \sum M_{Bi}. \quad (5.7)$$

Результати проведених розрахунків наведено у таблиці 5.3.

Таблиця 5.3 – Результати розрахунків матеріальних витрат

1	Найменування матеріальних ресурсів	Одиниця виміру	Фактично витрачено матеріалів	Ціна за одиницю, грн	Загальна сума витрат, грн
1	2	3	4	5	6
1. Основні матеріали					
1.1	Використання мережі Інтернет, місячна абонплата	міс		150	300,00
2. Допоміжні витрати					
2.1	Папір	уп.	0,2	85,00	17,00
2.2	Тонер	уп.	1	50,00	50,00
2.3	CD диск	шт.	2	10	20,00
Разом:					387,00

Загальні матеріальні витрати на Інтернет, папір формату А4, тонер та CD-диски становлять 387,00 грн.

5.4 Розрахунок витрат на електроенергію

Однією із статей витрат є витрати на електроенергію під час проходження усіх етапів реалізації кінцевого продукту. Затрати на електроенергію одиниці обладнання визначаються за формулою 5.8:

$$Z_e = W \cdot T \cdot S, \quad (5.8)$$

де W – необхідна потужність, кВт; T – кількість годин роботи обладнання; S – вартість кіловат-години електроенергії.

Вартість кіловат-години електроенергії слід приймати згідно існуючих на даний час тарифів. Отже, 1 кВт з ПДВ коштує 2,42 грн.

Потужність комп'ютерів для реалізації кінцевого продукту – 400 Вт, кількість годин роботи обладнання згідно таблиці 5.1 – 418 годин.

Визначимо витрати на електроенергію згідно формули 5.8

$$Z_e = 0,4 \cdot 418 \cdot 2,42 = 404,62 \text{ грн.}$$

Отже, затратами на електроенергію для методів та засобів реалізації програмних агентів для проектів класу «розумне місто» буде 404,62 грн.

5.5 Розрахунок суми амортизаційних відрахувань

Для будь якої діяльності характерною є властивість зношування на зниження якості властивостей інструментарію та фондів за допомогою яких ведеться діяльність. Для вирішення проблеми із відновленням даних фондів

використовується амортизація, що являє собою процес трансформації вартості основних фондів на вартість продукції, яка щойно була створена, задля повного відновлення основних фондів. Для визначення амортизаційних відрахувань використовується формула 5.9:

$$A = \frac{B_B \cdot H_A}{100 \%}, \quad (5.9)$$

де A – амортизаційні відрахування за звітний період, грн.;

B_B – балансова вартість групи основних фондів на початок звітного періоду, грн.;

H_A – норма амортизації, %.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %). Річний робочий фонд становитиме 2352 годин, так як робочий день становить 8 годин, а кількість робочих днів в місяці становить 24,5 годин.

Для даної розробки засобом розробки є комп'ютер. Його сума становить 18580 грн. Отже, амортизаційні відрахування будуть рівні:

$$A = 18580 \cdot 5 \% / 100 \% = 929,00 \text{ грн.}$$

Згідно проведених обчислень амортизаційні відрахування становлять 929,00 грн.

5.6 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління спілкою та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20-60 % від суми основної та додаткової заробітної плати працівників.

$$H_{\text{в}} = B_{\text{о.л.}} \cdot 0,2 \dots 0,6, \quad (5.10)$$

де $H_{\text{в}}$ – накладні витрати.

Отже, накладні витрати становлять згідно формули 5.10:

$$H_{\text{в}} = 62084,00 \cdot 0,2 = 12416,80 \text{ грн.}$$

Отже, накладні витрати для науково-дослідних робіт методів створення та використання онтологій в проектах «Розумних міст» будуть становити 12416,80 грн.

5.7 Складання кошторису витрат та визначення собівартості науково-дослідницької роботи

Результати проведених вище розрахунків зведемо у таблицю 5.4.

Таблиця 5.4 – Кошторис витрат на НДР

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	14421,00	59,3
Відрахування на соціальні заходи	3388,94	13,9
Матеріальні витрати	735	3,02
Витрати на електроенергію	321,024	1,32
Амортизаційні відрахування	2588,81	10,6
Накладні витрати	2884,20	11,9
Собівартість	24338,97	100

Собівартість (C_B) дослідження розрахуємо за формулою:

$$C_B = B_{o.n.} + B_{c.z.} + Z_{m.v.} + Z_e + A + H_g. \quad (5.11)$$

Отже, собівартість дослідження дорівнює:

$$C_B = 14421,00 + 3388,94 + 735 + 321,024 + 2588,81 + 2884,20 = 24338,97 \text{ грн.}$$

Загальний кошторис витрат та визначення собівартості науково-дослідницької роботи становить 24338,97 грн.

5.8 Розрахунок ціни проведених науково-дослідних робіт

Ціну науково-дослідної роботи можна визначити за формулою:

$$Ц = \frac{C_B \cdot (1 + P_{рен}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \quad (5.12)$$

де $P_{рен.}$ – рівень рентабельності, 30 %;

K – кількість замовлень, од.;

$B_{н.і.}$ – вартість носія інформації, грн.;

$ПДВ$ – ставка податку на додану вартість, (20 %).

Оскільки розробка є науково-дослідною, і використовуватиметься тільки один раз, то для розрахунку ціни не потрібно вказувати коефіцієнти K та $B_{н.і.}$, оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни розробки буде мати вигляд:

$$\Pi = C_B \cdot (1 + P_{pen}) \cdot (1 + ПДВ). \quad (5.13)$$

Звідси ціна проведення науково-дослідних робіт складе:

$$\Pi = 24338,97 (1 + 0,3) \cdot (1 + 0,2) = 37968,80 \text{ грн.}$$

Отже, для дослідження методів створення та використання онтологій в проектах «Розумних міст» необхідно 37968,80 грн.

5.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_p = \frac{\Pi}{C_B}, \quad (5.14)$$

де Π – прибуток;

C_B – собівартість.

Плановий прибуток ($\Pi_{пл}$) знаходимо за формулою:

$$\Pi_{пл} = \Pi - C_B. \quad (5.15)$$

Розраховуємо плановий прибуток:

$$\Pi_{пл} = 37968,80 - 24338,97 = 50854,25 \text{ грн.}$$

Отже, формула для визначення економічної ефективності набуде вигляду:

$$E_p = \frac{\Pi_{пл}}{C\vartheta}. \quad (5.16)$$

Тоді,

$$E_p = 13629,82 / 24338,97 = 0,56 .$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p}, \quad (5.17)$$

Термін окупності дорівнює:

$$T_p = 1 / 0,56 = 1,8 \text{ р.}$$

Згідно формул плановий прибуток від проведених науково-дослідних робіт становить 50854,25 грн., економічна ефективність дорівнює 0,56 , а термін окупності становить 1,8 роки що вважається доцільним та економічно вигідним.

5.10 Висновок

В розділі «Обґрунтування економічної ефективності» дипломної роботи освітнього рівня «магістр» розраховано основні техніко-економічні показники проведених досліджень методів та засобів реалізації програмних агентів для проектів класу «розумне місто» (див. таблицю 5.5).

Розраховане значення економічної ефективності становить 0,56 , що є високим значенням.

Так само нормальним є термін окупності, який повинен коливатися від 1 до 3 років. Для проведених в дипломній роботі досліджень він становить 1,8 років.

Таблиця 5.5 – Техніко-економічні показники НДР

№ п/п	Показник	Значення
1.	Собівартість, грн.	24338,97
2.	Плановий прибуток, грн.	13629,82
3.	Ціна, грн.	37968,80
4.	Економічна ефективність	0,56
5.	Термін окупності, рік	1,8

Отже, отримані в рамках дипломного проектування результати науково-дослідних робіт можуть бути впроваджені та мати подальший розвиток, оскільки вони є економічно вигідним за всіма основними техніко-економічними показниками.

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1 Покращення управління охорони праці в організації шляхом впровадження інформаційних технологій

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. З метою належного правового забезпечення необхідно розширити та доповнити перелік основних професій комп'ютерної галузі у національному класифікаторі ДК-003-2010, а також підготувати відповідний випуск у кваліфікаційному довіднику посад фахівців ІТ-індустрії, що сприятиме вирішенню питань їх соціального захисту, пенсійного забезпечення, атестації робочих місць основних професій за умовами праці на предмет подальших певних видів пільг та компенсацій за важкі шкідливі і небезпечні умови праці.

Важливим напрямом стосовно визначення професійної придатності фахівців з інформаційних технологій є проведення психофізіологічної експертизи відповідно до 5 статті Закону України «Про охорону праці».

Робота з комп'ютерами нового покоління характеризується певним психофізіологічними перенавантаженнями, втому зорового аналізатора, гіпокінезією, відсутність диференційованих норм праці при роботі з новою комп'ютерною технікою в залежності від віку, статі, категорії зорової роботи, режимів праці і відпочинку (протягом робочого дня, тижня, щорічного режиму відпусток).

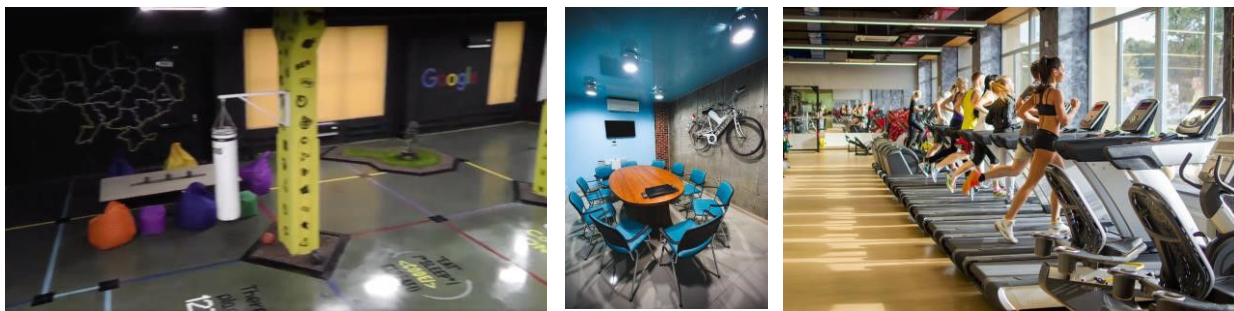
Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів.

Особлива роль з точки зору збереження та відновлення здоров'я працюючих в комп'ютерній галузі належить попереднім та періодичним наглядам з подальшої психофізіологічної експертизи і встановленням професійної придатності при роботі з комп'ютерами нового покоління, який супроводжується виникненням певних факторів професійного ризику електротравматизму при їх ремонті та обслуговуванні. В цьому зв'язку необхідне запровадження експертизи на предмет безпечної експлуатації ПЕОМ, тобто офіційне підтвердження фактичних параметрів електробезпеки, їх відповідності вимогам нормативної документації фахівців, які проводять таку експертизу повинні пройти навчання і перевірку знань відповідно до вимог ДНАОП 0.00-8.20-99. За результатами експертизи повинні прийматися рішення про відповідність ПЕОМ нормам безпеки, терміни чергової експертизи, оформлюються протоколи вимірювань і випробувань, проведені у разі потреби розрахунки та експертний висновок.

Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії.

Заслуговує на увагу зарубіжний досвід створення у приміщеннях та в зоні їх розміщення на територіях підприємств спеціальних візуальних комфортних умов та забезпечення вимог виробничої естетики, дотримання норм рівнів виробничого шуму та акустичної тиші за межами офісу (див рисунок 6.1).

Також дуже важливим є використання в офісних приміщеннях та кабінетах психофізіологічного розвантаження функціональної музики, яка сприяє попередженню перевтоми і підтриманню необхідного рівня розумової працездатності фахівців комп'ютерної галузі [54], [55].



а)

б)

в)

а – офіс для 20 програмістів ІТ-компанії; б – кімната для переговорів (мітінг-рум); в – спортивно-тренажерний комплекс компанії

Google (Великобританія)

Рисунок 6.1 – Приміщення сучасних ІТ-компаній за функціональним призначенням

В цьому напрямі заслуговує на увагу створення при великих центрах інформаційних технологій кімнат (кабінетів) психофізіологічного розвантаження працівників галузі (на 5 місць). Зарубіжний досвід охорони праці при використанні новітніх інформаційних технологій та сучасного комп'ютерного обладнання передбачає з метою попередження наслідків монотонної праці, підвищення рівня рухової активності і покращення розумової працездатності фахівців ІТ-індустрії під час технологічних перерв участь у спеціальних облаштованих приміщеннях необхідним спортивним інвентарем та різними тренажерами відповідних фізичних вправ, індивідуальних тренінгових завдань відповідно до віку, статі та категорії зорової роботи. Такий підхід дозволяє зняти надлишкове психофізіологічне перевантаження, підвищити працездатність центральної нервової системи, попередити перевтому зорового аналізатора. Показана ефективність проведення різноманітних за своєю спрямованістю вправ робітників цієї галузі (приблизно на 5-30%) [56].

Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства.

Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

6.2 Організаційно-технічні заходи та елементи системи протипожежного захисту робочих місць користувачів ПК

Організаційно-технічні заходи робочих місць користувачів ПК пов'язані як з системою попередження пожеж, так і з системою протипожежного захисту. До них відносяться:

- організація пожежної охорони;
- паспортизація речовин і матеріалів, технологічних процесів, будинків і споруд щодо забезпечення пожежної безпеки;
- залучення громадськості до забезпечення ПБ;
- організація навчання населення правилам пожежної безпеки;
- розробка і реалізація норм, правил, інструкцій про поведінку з пожежонебезпечними речовинами і матеріалами, про додержання протипожежного режиму, про дії людей при виникненні пожежі;
- виготовлення і застосування засобів агітації по забезпеченню пожежної безпеки; - порядок зберігання речовин і матеріалів в залежності від фізикохімічних і пожежонебезпечних властивостей, засобів їх гасіння;
- нормування чисельності людей на об'єкті за умови їх безпеки при пожежі;
- розробку заходів по діям адміністрації, робітників, службовців і населення на випадок виникнення пожежі, організація евакуації людей.

Основи протипожежного захисту підприємств визначені стандартами:

- ГОСТ 12.1. 004 - 76 "Пожежна безпека";
- ГОСТ 12.1.010 - 76 "Вибухобезпека. Загальні вимоги".

Цими ГОСТами можлива частота пожеж і вибухів допускається такої, щоб імовірність їх виникнення в течії року не перевищувала 10^{-6} або щоб імовірність впливу небезпечних чинників на людей в протягом року не перевищувала 10^{-6} на людину.

Заходи щодо пожежної профілактики розділяються на організаційні, технічні, режимні і експлуатаційні.

Організаційні заходи: передбачають правильну експлуатацію обладнання, правильний зміст будівель, території, протипожежний інструктаж і тому подібне.

Технічні заходи: дотримання протипожежних правил і норм при проектуванні будівель, при пристрої електропроводів і обладнання, опалювання, вентиляції, освітлення, правильне розміщення обладнання.

База нормативних документів, які регулюють вимоги до систем протипожежного захисту складається з наступного:

- Вимоги до систем протипожежного захисту визначаються Державними Будівельними Нормами України (ДБН В.2.5-56:2014 [57]).

- Постанова про затвердження Правил пожежної безпеки в Україні від 30 грудня 2014 № 1417.

Ці документи визначають склад конкретних систем протипожежного захисту (далі – СПЗ) та вимоги до них. До таких систем належать:

- Системи пожежної сигналізації.
- Автоматичні системи пожежогасіння.
- Автономні системи пожежогасіння локального застосування.
- Системи оповіщення про пожежу та управління евакуюванням людей.
- Системи протидимного захисту.
- Системи централізованого пожежного спостерігання.
- Системи диспетчеризації СПЗ.

6.3 Забезпечення електробезпеки користувачів ПК

Приміщення із робочими місцями користувачів комп'ютерів для забезпечення електробезпеки обладнання, а також для захисту від ураження електричним струмом самих користувачів ПК повинні мати достатні технічні засоби захисту.

Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливості, перейти на негорючу ізоляцію.

Лінія електромережі для живлення ЕОМ, периферійних пристроїв ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ виконується як окрема групова трипровідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. [58]

Використання нульового робочого провідника як нульового захисного провідника забороняється. Нульовий захисний провід прокладається від стійки групового розподільного щита, розподільного пункту до розеток живлення. Не допускається підключення на щиті до одного контактного затискача нульового робочого та нульового захисного провідників. Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі повинна бути не менше площі перерізу фазового провідника.

Усі провідники повинні відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам Правил налаштування електроустановок.

У приміщенні, де одночасно експлуатується або обслуговується більше п'яти персональних ЕОМ, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ повинні підключатися до електромережі тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника. [59]

Неприпустимим є підключення ЕОМ та периферійних пристроїв ЕОМ до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.

Електромережі штепсельних з'єднань та електророзеток для живлення ЕОМ, периферійних пристроїв слід виконувати за магістральною схемою, по 3...6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 36 В за своєю конструкцією повинні відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В і мають бути пофарбовані в колір, який візуально значно відрізняється від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

Індивідуальні та групові штепсельні з'єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах з урахуванням вимог Правил налаштування електроустанов та Правил пожежної безпеки в Україні.

Електромережу штепсельних розеток для живлення ЕОМ, периферійних пристроїв ЕОМ при розташуванні їх уздовж стін приміщення прокладають по підлозі поряд зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання.

При розташуванні в приміщенні за його периметром до 5 ЕОМ, використанні трипровідникового захищеного проводу або кабелю в оболонці з негорючого або важкогорючого матеріалу дозволяється прокладання їх без металевих труб та гнучких металевих рукавів.

Електромережу штепсельних розеток для живлення ЕОМ при розташуванні їх у центрі приміщення, прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах. При цьому не дозволяється застосовувати провід і кабель в ізоляції з вулканізованої гуми та інші матеріали, що містять сірку. Відкрита прокладка кабелів під підлогою забороняється. Металеві труби та гнучкі металеві рукави повинні бути заземлені. Заземлення повинно відповідати вимогам НПАОП 40.1-1.21-98.

Для підключення переносної електроапаратури застосовують гнучкі проводи в надійній ізоляції.

Тимчасова електропроводка від переносних приладів до джерел живлення виконується найкоротшим шляхом без заплутування проводів у конструкціях машин, приладів та меблях. Доточувати проводи можна тільки шляхом паяння з наступним старанним ізолюванням місць з'єднання.

Є неприпустимими:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією; залишення під напругою кабелів та проводів з неізолюваними провідниками;

- застосування саморобних подовжувачів, які не відповідають вимогам Правил влаштування електроустанов до переносних електропроводок;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарювання;
- користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;
- підвішування світильників безпосередньо на струмопровідних проводах, обгортання електроламп і світильників папером, тканиною та іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);
- використання електроапаратури та приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

6.4 Оцінка дії радіоактивного забруднення місцевості після ядерного вибуху на виробничу діяльність муніципальних підприємств, установ та організацій

Виявлення радіаційної обстановки методом прогнозування – це перший етап роботи. У цій оцінці використовують інформацію про ядерні вибухи і дані про напрямки і швидкості вітру. Така оцінка дає можливість орієнтовно визначити вплив радіоактивного забруднення місцевості на дієздатність рятувальних формувань, можливість функціонування об'єктів муніципального господарства, вибрати найбільш доцільні способи дій на забрудненій місцевості, намітити заходи протирадіаційного захисту, а також дати завдання для ведення радіаційної розвідки. Радіуси зон забруднення в районі наземного вибуху з навітряної сторони наведені в таблиці 6.1.

Таблиця 6.1 – Радіуси зон забруднення в районі наземного вибуху з навітряної сторони, м

Потужність вибуху, Мт	Зони забруднення			Потужність вибуху, Мт	Зони забруднення		
	А	В	В		А	Б	В
0,02	735	450	340	0,2	1070	735	595
0,05	865	560	430	0,5	1220	865	710
0,10	970	645	510	1,0	1290	930	770

Розміри зони радіоактивного забруднення при ядерних вибухах різної потужності і швидкостях середнього вітру наведені в таблиці 6.2.

Таблиця 6.2 – Розміри зони радіоактивного забруднення при ядерних вибухах різної потужності і швидкостях середнього вітру

Потужність вибуху, Мт	Швидкість середнього вітру, км/год	Розміри зон і еталонний рівень радіації, км		
		А – 8 Р/год	Б – 80 Р/год	В – 240 Р/год
1	2	3	4	5
0,1	25	116–12	49–6,1	31–4
	50	150–14	60–6,4	35–3,9
	75	175–15	64–6,3	35–3,8
0,2	25	157–15	67–7,8	43–5,3
	50	200–18	83–8,3	50–5,3
	75	233–20	90–8,4	50–5,0
0,3	25	190–18	80–8,0	52–6,0
	50	240–21	98–9,6	60–6,2
	75	275–23	100–9,8	60–6,0
0,5	25	231–21	100–10	65–7,4
	50	300–25	121–12	78–7,7
	75	340–27	140–13,5	83–7,7
1,0	25	309–20	132–13,0	83–9,5
	50	402–31	170–15,0	109–10,0
	75	466–34	192–16,0	118–10,0
2,0	25	413–32	182–17	121–12
	50	536–39	231–19,6	149–13
	75	626–43	262–21	165–13

Продовження таблиці 6.2

1	2	3	4	5
3,0	25	495–37	218–19	145–14
	50	630–45	275–23	180–15
	75	750–50	310–24	200–16
5,0	25	772–52	343–27	225–19
	50	920–58	393–28	253–20
	75	1035–62	436–30	270–20

* Перша група чисел – довжина зони, друга – ширина.

Оцінка проводиться в такій послідовності: визначають розміри зон радіоактивного забруднення; наносять на карту (схему) зони радіоактивного забруднення; розраховують час випадання радіоактивних речовин.

Розміри зон радіоактивного забруднення визначають за допомогою таблиць, радіаційних і розрахункових лінійок.

При нанесенні на карту (схему) зон радіоактивного забруднення спочатку накреслюють центр ядерного вибуху. Зверху, зліва від нього, записують: у чисельнику – вид вибуху і потужність, у знаменнику – час і дату вибуху. Користуючись даними довідкових таблиць, потрібно нанести межу зони забруднення в районі вибуху, враховуючи його потужність. Потім від центру вибуху провести пряму лінію – вісь сліду, що відповідає напрямку руху середнього вітру. Відкласти довжину і ширину кожної зони забруднення за даними табл. 36. Від кола зони забруднення, враховуючи ширину і довжину нанести зони, кожна певного кольору: зона А – синя, зона Б – зелена, зона В – коричнева, зона Г – чорна. Зони позначають з урахуванням масштабу карти (див. рисунок 6.2).

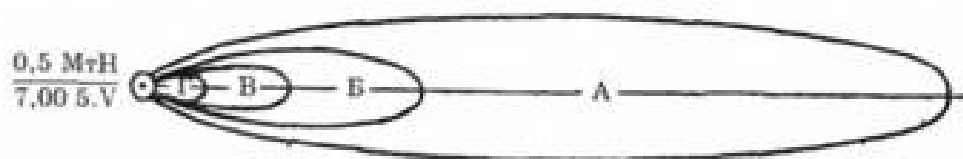


Рисунок 6.2 – Схема прогнозу зон радіоактивного забруднення ядерного вибуху

Час випадання радіоактивних речовин визначають за формулою:

$$t_{\text{вип}} = R/v,$$

де R – відстань від центру вибуху до даного об'єкта або населеного пункту, км;

v – швидкість середнього вітру, км/год.

Другий етап роботи – це виявлення фактичної радіаційної обстановки та її оцінка. На основі даних, одержаних від радіаційної розвідки, орган управління цивільного захисту і командири формувань оцінюють радіаційну обстановку.

Радіаційна обстановка визначається безпосередньо на об'єкті, навколо нього, на маршрутах висування сил цивільного захисту, а також у районі евакуації, уточнюється імовірний час початку випадання радіоактивних речовин.

Радіаційна обстановка характеризується масштабами і характером радіоактивного забруднення. Основними показниками ступеня небезпеки радіоактивного забруднення для населення є розміри зон радіоактивного забруднення і рівні радіації. Проте оцінюючи вплив радіоактивного забруднення на дії рятувальних формувань цивільного захисту і життєдіяльність населення, необхідно обов'язково враховувати і ступінь захищеності людей від радіоактивних випромінювань.

Оцінка радіаційної обстановки – це розв'язання основних завдань різних варіантів дій рятувальних служб цивільного захисту, а також виробничої діяльності об'єктів і галузей виробництва в умовах радіоактивного забруднення, аналіз одержаних результатів і вибір найбільш доцільних варіантів дій, які б виключали радіаційне ураження людей, сільськогосподарських тварин і забруднення радіоізотопами урожаю, продуктів і води.

Для оцінки радіаційної обстановки в населеному пункті й на виробничому об'єкті за даними розвідки необхідні такі вихідні дані.

1. Час ядерного вибуху, від якого виникло радіоактивне забруднення. Ці дані можна одержати з відділу питань НС та цивільного захисту населення району, області або методом розрахунку.

2. Рівні радіації на об'єкті та час їх випромінювання. Через те, що заміри рівнів радіації на об'єкті проводяться неодноразово, доцільно під час оцінки радіаційної обстановки значення рівнів радіації привести до 1 год після ядерного вибуху.

3. Значення коефіцієнтів ослаблення радіації будовами, спорудами, сховищами, укриттями, транспортними засобами. Для цього можна використати середні значення коефіцієнтів ослаблення, одержані розрахунково. Але надійніше після випадання радіоактивних речовин уточнити ці коефіцієнти замірюванням рівнів радіації всередині будинку (споруди), де будуть знаходитися люди, і на відкритій місцевості на відстані 20–30 м від будинку (споруди):

$$K_{\text{осл}} = P_{\text{відкр}}/P_{\text{буд}},$$

де $P_{\text{відкр}}$ – рівень радіації на відкритій місцевості;

$P_{\text{буд}}$ – рівень радіації в будинку (споруді).

Інтервал між двома замірюваннями не повинен перебільшувати 2–3 хв.

4. Допустимі дози опромінення встановлюють залежно від конкретної обстановки, характеру завдання, яке будуть виконувати рятувальні формування. Необхідно враховувати, яке опромінення може бути одержане – одноразове чи багаторазове.

Слід враховувати те, що спочатку накопичення дози опромінення відбувається інтенсивніше, тому встановлену дозу перші чотири доби необхідно ділити у відповідній пропорції.

Кінцевим етапом оцінки радіаційної обстановки є висновки начальника цивільного захисту об'єкта про вплив радіоактивного забруднення на виробничу діяльність об'єкта, ведення рятувальних і невідкладних робіт на об'єкті, проведення виробничих робіт; найбільш доцільний варіант дій формувань при перетинанні зон і веденні рятувальних робіт на об'єкті:

- заходи захисту населення і особового складу формувань;
- кому і які необхідно дати розпорядження з метою забезпечення дій формувань в умовах радіоактивного забруднення;
- заходи захисту населення, тварин, рослин, врожаю, продуктів, сировини, кормів, вододжерел;
- визначення сил і засобів для ліквідації наслідків радіоактивного забруднення;
- заявка для завезення необхідних засобів.

Висновки з оцінки радіаційної обстановки знаходять відображення в рішенні начальника цивільного захисту для організації рятувальних і невідкладних робіт і є основою для організації захисту особового складу рятувальних формувань і населення в умовах радіоактивного забруднення.

Оцінюючи обстановку, можна користуватись формулами, спеціальними таблицями, графіками, лінійками: дозиметричною (ДЛ), радіаційними (РЛ-І, РЛ-3), розрахунковою лінійкою, обчислювальною технікою.

6.5 Висновок

В розділі описано покращення управління охорони праці в організації шляхом впровадження інформаційних технологій. Зокрема підкреслено важливість створення при великих центрах інформаційних технологій кімнат психофізіологічного розвантаження працівників галузі.

Окремо подано організаційно-технічні заходи та елементи системи протипожежного захисту робочих місць користувачів ПК. Зокрема наведено їх класифікацію відповідно до Державних Будівельних Норм України.

Також описано забезпечення електробезпеки користувачів ПК та оцінка дії радіоактивного забруднення місцевості після ядерного вибуху на виробничу діяльність муніципальних підприємств, установ та організацій.

7 ЕКОЛОГІЯ

7.1 Екологічна ситуація в Україні

В Україні за двадцять вісім років незалежності питання екології піднімалося безліч разів та навіть декларувалося на найвищому державному рівні. На жаль даліше обговорень та розроблень планів з реалізації, хоча б поступової не дійшло, по вирішенню екологічної проблеми забруднення навколишнього середовища. Сучасна епоха – це епоха бурхливого розвитку науки і техніки, неконтрольованого зростання населення Землі, поступової деградації природного середовища під впливом негативних антропогенних чинників. В сучасних умовах швидкість науково – технічного прогресу на кілька порядків перевищує адаптаційні можливості існуючих живих організмів. Тому надзвичайно загострились багато проблем, а серед них одна з головних – проблема взаємозв'язку суспільства і природи, людини і навколишнього середовища. Все це призводить до порушення екологічної рівноваги, що складалася протягом тривалого часу і спричинює появу екологічної кризи, небезпечної для людей і довкілля. [60]

Зростаючий антропогенний вплив на навколишнє середовище, його забруднення різними відходами виробництва, поряд з надмірним використання природних ресурсів, стали предметом широкого обговорення і всестороннього вивчення. Саме тому метою нашої роботи є дослідження сучасних екологічних проблем України і пошук шляхів гармонізації відносин природи і суспільства. Найбільший антропогенний вплив на навколишнє середовище в сучасну епоху чинить транспорт, промисловість, енергетика, сільське господарство.

Найочевиднішим прикладом є стан Дніпра в міській смузі столиці: головна водна артерія в буквальному сенсі перетворюється на болото. Для рівнян проблема забруднення міських водойм своїми власними діями,

найбільше знайома. Скрізь можна спостерігати одну й ту саму тенденцію – екологічні проблеми накопичуються та консервуються. А чиновники незмінно оперують тезою про відсутність належних коштів в держбюджеті/місцевих бюджетах на вирішення цих проблем.

Другою важливою проблемою забруднення є транспорт, що належить до головних забруднювачів атмосферного повітря, водойм і ґрунтів. Відбувається деградація екосистем під впливом транспортних забруднень, особливо інтенсивно спостерігається у містах. Транспорт належить до головних забруднювачів атмосферного повітря, водоймищ і ґрунтів. Відбувається деградація екосистем під впливом транспортних забруднень, особливо інтенсивно на урбанізованих територіях. Гостро стоїть проблема утилізації і переробки відходів, що з'являються при експлуатації транспортних засобів. Для потреб транспорту у великій кількості споживаються природні ресурси. Вихлопні гази автомобілів містять більш ніж 200 хімічних сполук-продуктів згорання палива, більшість з яких токсичні. Особливо гостро постає проблема аварій крупнотонажних вантажних суден, які здійснюють перевезення нафтопродуктів. Такі аварії завдають велику шкоду водному середовищу і викликали вже не одну екологічну катастрофу [61]. Гостро стоїть проблема утилізації і переробки відходів, що з'являються при експлуатації транспортних засобів. Для потреб транспорту у великій кількості споживаються природні ресурси. Вихлопні гази автомобілів містять до двохсот хімічних сполук згорання палива, більшість з яких токсичні.

У разі подальшого зберігання подібного стану речей наша держава ризикує скотитися в плані екологічної небезпеки для життя людей на рівень африканських країн. Тому ключове завдання громадськості – тримати основні екологічні проблеми в фокусі суспільної уваги, добиватися від керівництва держави реального покращення ситуації з охороною навколишнього середовища.

Не менш небезпечним забруднювачем оточуючого середовища є промислові відходи. В Україні основним джерелом утворення відходів є підприємства гірничо-промислового, хіміко-металургійного, машинобудівного, паливно-енергетичного, будівельного, агропромислового комплексів, а також комунальне господарство. Найтоксичнішими серед них є відходи, що містять важкі метали, нафтопродукти, непридатні для застосування отрутохімікати (пестициди), основна маса яких утворюється в Донецькій та Дніпропетровській областях. Під сховищами токсичних відходів перебуває майже 20 тис.га земель. Це сміття звозиться на звалища, переважна більшість яких є джерелом інтенсивного забруднення води і повітря [62].

Як свідчать статистичні данні, приблизно 80% всіх видів забруднення повітря – наслідок енергетичних процесів (добування, переробка й використання енергоресурсів). Особливо шкідливими є викиди сполук сірки в районах металургійних заводів. Сполучаючись з парами води в атмосфері, триоксид сірки утворює сірчану кислоту, суспензії якої є дуже небезпечними. У викидах ТЕЦ небезпечні також оксиди важких металів, фтористі сполуки, бензоперен, що відносяться до канцерогенних речовин. АЕС разом із електроенергією виробляють велику кількість надзвичайно небезпечних речовин [63].

Основними негативними наслідками сільськогосподарської діяльності людини є збідніння і виснаження родючих українських чорноземів, промислове забруднення ґрунтів та інтенсивне освоєння цілинних земель, широке розповсюдження монокультур, застосування азотних і нітратних мінеральних добрив [64].

Всі вище зазначенні чинники згубно впливають на екологічну ситуацію в Україні і викликають такі негативні наслідки як парниковий ефект, кислотні дощі, руйнування озонового шару.

Озоновий шар Землі – це верхній найтонший (близько 3мм) шар атмосфери, що знаходиться на висоті від 10-17 до 50 км (в залежності від широти і пори року). Він виконує одну з найважливіших функцій на нашій планеті – захищає від ультрафіолетового випромінювання, яке згубно впливає на всі живі організми. Глобальні зміни клімату і господарча діяльність людини призводять до стоншення озонового шару, що впливає на рівень інтенсивності ультрафіолетового випромінювання Сонця. Ще одною серйозною проблемою сучасності є підвищення концентрації озону біля земної поверхні, тому що у великій кількості він отруйний для живих організмів. Зокрема, у рослинах він руйнує клітинні мембрани листків, що призводить до вимирання лісів, у людей і тварин – знижує опірність організму інфекційним захворюванням легень і дихальних шляхів. Основною причиною дисбалансу між вмістом озону в атмосфері і біля поверхні Землі є викиди забруднюючих речовин в атмосферу: вихлопні гази автомобілів, хлорфторвуглеводні (холодильна техніка, аерозолі), хімічні добрива, продукти згоряння промислового пального, викиди відпрацьованих газів при запуску ракет і висотних літаків, атомні вибухи [65].

Наступною глобальною екологічною проблемою нашого часу, що потребує негайного вирішення є проблема парникового ефекту. За останні 100 років концентрація в атмосфері вуглекислого газу збільшилась на 25%, а метану – на 100%. Ці гази зумовлюють виникнення так званого «парникового ефекту»: пропускаючи сонячне світло, вони частково затримують теплове випромінювання, яке надходить від поверхні Землі. Цей процес супроводжується глобальним підвищенням температури приблизно на 0,5°C. У найближчий час викиди в атмосферу метану та вуглекислого газу будуть зростати, з чого більшість вчених роблять висновок, що через парниковий ефект протягом 50 років середня температура на Землі може підвищитися майже на 2-5°C. Це потепління може викликати підвищення рівня Світового океану на 0,5 – 1,5м , що призведе до затоплення густонаселених

прибережних районів. Підвищення планетарної температури створює умови для відходження криги з полюсів на Північ, що призведе до їх розтавання [66].

В таких умовах бездумної експлуатації багатств природи, активної неконтрольованої діяльності людини, виникає питання про існування самого людства, оскільки існує реальна загроза його знищення. Люди поставили себе над природою, забувши, що вони є її частиною і підкоряються її законам. Тож для успішного розв'язання складних екологічних проблем насамперед потрібен перехід до утвердження екологічної свідомості як на індивідуальному, так і на суспільному рівні. Природні ресурси і довкілля як загальнолюдську спадщину треба використовувати з дотриманням принципу рівних прав нинішнього і майбутніх поколінь на доброякісне природне життєве довкілля і достатню кількість і якість природних ресурсів [67].

7.2 Вимоги до моніторів (ВДТ) і ПЕОМ

Чіпкість зображення моніторів, в першу чергу залежать від роздільної здатності монітора, яка визначається числом дискретних елементів зображення, відтворених монітором по горизонталі і вертикалі. Чим вище роздільна здатність, тим точніше і чіпкіше зображення на екрані, тим легше воно для сприйняття, тим менше стомлює зорову систему. При низькому розрізненні можливі помилки при прочитуванні символів (два різні символи при малій кількості складових можуть сприйматися як однакові). Існують стандартні значення роздільної здатності (у дужках наведена назва стандарту для ПК): 640x480 (VGA); 800x600 (SVGA); 1024x768 (XGA); 1280x1024 (EVGA); 1600x1200 (не позначений) – максимальне значення для сучасних моніторів.

Чіпкість зображення залежить, крім того, від кроку люмінофора (dot pitch) – відстані між дискретними точками люмінофора одного кольору на

внутрішній поверхні екрана. Для апертурної градки – це відстань між смугами одного кольору, для тіньової маски лінія мінімальної відстані між точками одного кольору становить з горизонталлю кут 30° . У різних моделей моніторів крок люмінофора лежить в діапазоні від 0,25 до 0,41 мм. Для інтенсивних робіт з графікою при розрізненні вище 1024x768 переважно крок 0,25 або 0,26 мм. Для ділового і домашнього застосування в більшості додатків, що використовують режим розрізнення 1024x768 або нижче, достатній крок 0,27 або 0,28 мм. В нормах встановлені такі вимоги до ВДТ (див. таблицю 7.1).

Таблиця 7.1 – Вимоги до відеотерміналів

Найменування параметра	Значення параметра
Яскравість знака (яскравість фону), кд/кв.м	від 35 до 120
Зовнішня освітленість екрана, лк	від 100 до 250
Контраст (для монохромних зображень)	від 3 : 1 до 1,5 : 1
Нерівномірність яскравості в робочій зоні екрана	не більше 1,7 : 1
Відхилення форми робочої зони екрана від прямокутності по горизонталі і вертикалі	не більше 2%
Різниця довжин рядків або стовпчиків	не більше 2% середнього значення
Розмір мінімального елементу зображення (пікселя) для монохромних зображень, мм	0,3
Допустима тимчасова нестабільність зображення (мерехтіння)	не повинна бути зафіксована у 90 відсотків спостерігачів
Відбивна властивість, дзеркальне і змішане віддзеркалення (відблиск) %, (допускається виконання вимог при застосуванні приєкранного фільтра)	не більше 1
Відношення ширини знака до його висоти для великих літер	від 0,7 до 0,9
Мінливість розміру знака	не більше 5% висоти
Ширина лінії контура знака	0,15...0,1 висоти знака
Модуляція щодо яскравості растру: для багатоколірних зображень	не більше 0,7

Велике значення для чіткості зображення має якість фокусування електронних променів.

7.3 Висновки до розділу

В розділі екологія магістерської роботи проаналізовано екологічну ситуацію сформовану на даний час в Україні. Відзначено роль антропогенного впливу на навколишнє середовище. Також виділено транспорт, як один з головних забруднювачів атмосферного повітря, водойм і ґрунтів. Окрім того виділено ще один не менш небезпечний забруднювач оточуючого середовища – промислові відходи.

В розділі подано та проаналізовано вимоги до моніторів (ВДТ) і ПЕОМ.

ВИСНОВКИ

В процесі виконання дипломної роботи освітнього рівня «магістр» проведено аналіз предметної області, зокрема:

- досліджено середовище сучасних проектів «розумних міст»;
- проаналізовано існуючі програмні агенти в системах «розумного міста»;
- досліджено та описано інструментарій для створення програмних агентів в «розумному» міському середовищі;
- розглянуто мови програмування і програмні платформи для створення «розумних» програмних агентів.

В другому розділі дипломної роботи:

- досліджено засоби специфікації моделей «розумних» програмних агентів;
- проаналізовано інструментальні засоби для розроблення та побудови «розумних» програмних агентів;
- наведено порівняльний аналіз інструментальних середовищ.

В третьому розділі дипломної роботи:

- запропонована архітектура мультиагентних застосунків «розумного міста»;
- розроблена модель ресурсів «розумного» програмного агента;
- наведено опис розробленого програмного модуля «розумного» міського програмного агента.

В розділі «Спеціальна частина» відображена реалізація моделі міського «розумного» програмного агента.

В розділі «Обґрунтування економічної ефективності» розраховано основні техніко-економічні показники проведених досліджень.

В розділі «Охорона праці та безпека в надзвичайних ситуаціях» описано покращення управління охорони праці в організації шляхом

впровадження інформаційних технологій та досліджено організаційно-технічні заходи та елементи системи протипожежного захисту робочих місць користувачів ПК. Описано забезпечення електробезпеки користувачів ПК та оцінка дії радіоактивного забруднення місцевості після ядерного вибуху на виробничу діяльність муніципальних підприємств, установ та організацій.

В розділі «Екологія» описана екологічна ситуація в Україні та подано вимоги до моніторів (ВДТ) і ПЕОМ.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Tabachyshyn, Danylo, et al. "Information Systems for Processes Maintenance in Socio-communication and Resource Networks of the Smart Cities." *Conference on Computer Science and Information Technologies*. Springer, Cham, 2018.
- 2 Liu, Yi, et al. "Intelligent edge computing for IoT-based energy management in smart cities." *IEEE Network* 33.2 (2019): 111-117.
- 3 Osman, Ahmed M. Shahat. "A novel big data analytics framework for smart cities." *Future Generation Computer Systems* 91 (2019): 620-633.
- 4 Kakderi, Christina, Nicos Komninos, and Panagiotis Tsarchopoulos. "Smart cities and cloud computing: lessons from the STORM CLOUDS experiment." *Journal of Smart Cities* 1.2 (2019): 4-13.
- 5 Ahuja, Kiran, and Arun Khosla. "Network selection criterion for ubiquitous communication provisioning in smart cities for smart energy system." *Journal of Network and Computer Applications* 127 (2019): 82-91.
- 6 Fong, Sim Liew, et al. "Smart City Bus Application with Quick Response (QR) Code Payment." *Proceedings of the 2019 8th International Conference on Software and Computer Applications*. ACM, 2019.
- 7 Duda, Oleksii, et al. "Cloud-based IT Infrastructure for “Smart City” Projects." *Dependable IoT for Human and Industry: Modeling, Architecting, Implementation* (2018): 389.
- 8 Cardullo, Paolo, and Rob Kitchin. "Being a ‘citizen’ in the smart city: up and down the scaffold of smart citizen participation in Dublin, Ireland." *GeoJournal* 84.1 (2019): 1-13.
- 9 Naranjo, Paola G. Vinueza, et al. "FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments." *Journal of Parallel and Distributed Computing* 132 (2019): 274-283.

-
- 10 Julian, Vicente, and Vicente Botti. "Multi-Agent Systems." (2019): 1402.
 - 11 Rath, Mamata, Bibudhendu Pati, and Binod Kumar Pattanayak. "Mobile agent-based improved traffic control system in VANET." *Integrated Intelligent Computing, Communication and Security*. Springer, Singapore, 2019. 261-269.
 - 12 Liu, Yi, et al. "Intelligent edge computing for IoT-based energy management in smart cities." *IEEE Network* 33.2 (2019): 111-117.
 - 13 Del Esposte, Arthur de M., et al. "Design and evaluation of a scalable smart city software platform with large-scale simulations." *Future Generation Computer Systems* 93 (2019): 427-441.
 - 14 Olszewski, Robert, et al. "Spatiotemporal Modeling of the Smart City Residents' Activity with Multi-Agent Systems." *Applied Sciences* 9.10 (2019): 2059.
 - 15 Savaglio, Claudio, et al. "Agent-based Internet of Things: State-of-the-art and research challenges." *Future Generation Computer Systems* 102 (2020): 1038-1053.
 - 16 Kapitonov, Aleksandr, et al. "Robotic Services for New Paradigm Smart Cities Based on Decentralized Technologies." *Ledger* 4 (2019).
 - 17 Olszewski, Robert, et al. "Spatiotemporal Modeling of the Smart City Residents' Activity with Multi-Agent Systems." *Applied Sciences* 9.10 (2019): 2059.
 - 18 Savaglio, Claudio, et al. "Agent-based Internet of Things: State-of-the-art and research challenges." *Future Generation Computer Systems* 102 (2020): 1038-1053.
 - 19 Garnett, Philip. "Growing Smart Cities." *From Astrophysics to Unconventional Computation*. Springer, Cham, 2020. 299-310.
 - 20 Sakurada, Lucas Hiroaki. *Development of industrial agents in a smart parking system for bicycles*. Diss. 2019.

-
- 21 Merabet, Ghezlane Halhoul, Mohamed Essaaïdi, and Driss Benhaddou. "Contributing to the Modeling of a Multi-Agent System for Comfort Control in Smart Buildings Applications." *International Journal of Computer Applications* 975: 8887.
 - 22 Tazi, Khadija, Fouad Mohamed Abbou, and Farid Abdi. "Multi-agent system for microgrids: design, optimization and performance." *Artificial Intelligence Review* (2019): 1-60.
 - 23 Ferrando, Angelo, and Theses Series DIBRIS-TH. "An Unexpected Journey: Towards Runtime Verification of Multiagent Systems and Beyond."
 - 24 Mohebalı, Behshad, et al. "A big data inspired preprocessing scheme for bandwidth use optimization in smart cities applications using Raspberry Pi." *Big Data: Learning, Analytics, and Applications*. Vol. 10989. International Society for Optics and Photonics, 2019.
 - 25 Mackay, Jacob, Graham Brooker, and David Johnson. "Unique Road Features for Localisation and Navigation Illuminated by 94 GHz Millimetre Wave Radar."
 - 26 Sharifi, Ayyoob. "A critical review of selected smart city assessment tools and indicator sets." *Journal of Cleaner Production* (2019).
 - 27 Spagnoli, Francesca, Shenja van der Graaf, and Martin Brynskov. "The Paradigm Shift of Living Labs in Service Co-creation for Smart Cities: SynchroniCity Validation." *Organizing for Digital Innovation*. Springer, Cham, 2019. 135-147.
 - 28 Rodrigues, Henrique Donâncio Nunes. *Avaliação de escalabilidade e desempenho da camada de transporte de mensagens em plataformas multiagente*. Diss. Universidade de São Paulo, 2019.
 - 29 Bunders, Damion J., and Krisztina Varró. "Problematizing data-driven urban practices: Insights from five Dutch 'smart cities'." *Cities* 93 (2019): 145-152.

30 Nesti, Giorgia. "Mainstreaming gender equality in smart cities: Theoretical, methodological and empirical challenges." *Information Polity* 24.3 (2019): 289-304.

31 Mora, Luca, Mark Deakin, and Alasdair Reid. "Combining co-citation clustering and text-based analysis to reveal the main development paths of smart cities." *Technological Forecasting and Social Change* 142 (2019): 56-69.

32 Criado, J. Ignacio, and J. Ramon Gil-Garcia. "Creating public value through smart technologies and strategies." *International Journal of Public Sector Management* (2019).

33 Samouylov, Konstantin, Evgeny Popov, and Konstantin Semyachkov. "Institutional Support of a Smart City." *Montenegrin Journal of Economics* 15.4 (2019): 87-98.

34 Guerard, Guillaume, and Hugo Pousseur. "JADE Modeling for Generic Microgrids." *Agents and Multi-agent Systems: Technologies and Applications 2019*. Springer, Singapore, 2020. 377-386.

35 Karpinski, Mykolay, Oleksiy Duda, and Oleksandr Matsiuk. "Information Systems for Processes Maintenance in Socio-communication and Resource Networks of the Smart Cities." *Advances in Intelligent Systems and Computing III: Selected Papers from the International Conference on Computer Science and Information Technologies, CSIT 2018, September 11-14, Lviv, Ukraine*. Vol. 871. Springer, 2018.

36 Stoyanov, Stanimir, et al. "A Reference Architecture Supporting Smart City Applications." *International Conference on Business Information Systems*. Springer, Cham, 2019.

37 Abbas, Hosny, Samir Shaheen, and Mohammed Amin. "Engineering Large Complex Critical Infrastructures of Future Smart Cities as Self-adaptive Systems." *Security in Smart Cities: Models, Applications, and Challenges*. Springer, Cham, 2019. 143-170.

38 Balicki, Jerzy, Piotr Dryja, and Marcin Zakidalski. "Some Artificial Intelligence Driven Algorithms For Mobile Edge Computing in Smart City." *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, Cham, 2019.

39 Attoh, Kafui, Katie Wells, and Declan Cullen. "'We're building their data': Labor, alienation, and idiocy in the smart city." *Environment and Planning D: Society and Space* (2019): 0263775819856626.

40 Mualla, Yazan, et al. "Between the megalopolis and the deep blue sky: Challenges of transport with uavs in future smart cities." *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

41 Tanda, Adriano, and Alberto De Marco. "Business Model Framework for Smart City Mobility Projects." *IOP Conference Series: Materials Science and Engineering*. Vol. 471. No. 9. IOP Publishing, 2019.

42 Urrutia-Azcona, Koldo, et al. "Smart Zero Carbon City: key factors towards smart urban decarbonisation." *DYNA* 94.6 (2019): 676-683.

43 Đurić, B. Okreša, et al. "MAMbO5: a new ontology approach for modelling and managing intelligent virtual environments based on multi-agent systems." *Journal of Ambient Intelligence and Humanized Computing* 10.9 (2019): 3629-3641.

44 Ranjan, Rajeev, et al. "Performance Evaluation of Sustainable Smart Cities in India: An Adaptation of Cartography in PROMETHEE-GIS Approach." *Advanced Multi-Criteria Decision Making for Addressing Complex Sustainability Issues*. IGI Global, 2019. 14-40.

45 Qasim, Awais, et al. "Timed-Arc Petri-Nets based Agent Communication for Real-Time Multi-Agent Systems."

46 Alsaig, Alaa, et al. "Characterization and Efficient Management of Big Data in IoT-Driven Smart City Development." *Sensors* 19.11 (2019): 2430.

47 JothiJanifer, N. A., and B. Gunasundari. "Ambulance Service Using Android Application."

48 Marie-Noëlle Brisson, C. R. E., Dan Doggendorf, and Michael Savoie. "Cybersecurity of Building Technology: Smart Cities and Smart Buildings Require Smart Protection." (2019).

49 Stoyanov, Stanimir, et al. "A Reference Architecture Supporting Smart City Applications." *International Conference on Business Information Systems*. Springer, Cham, 2019.

50 Mualla, Yazan, et al. "Towards Explainability for a Civilian UAV Fleet Management using an Agent-based Approach." *arXiv preprint arXiv:1909.10090* (2019).

51 Mustafa, Ola, et al. "Applying GranMicro Approach for Migrating Smart Cities Applications to Microservices." *Smart Cities/Smart Regions–Technische, wirtschaftliche und gesellschaftliche Innovationen*. Springer Vieweg, Wiesbaden, 2019. 529-538.

52 Panisson, Alison Roberto. "A framework for reasoning and dialogue in multi-agent systems using argumentation schemes." (2019).

53 Nelson, Alexander, et al. "Replication of Smart-City, Internet of Things Assets in a Municipal Deployment." *IEEE Internet of Things Journal* (2019).

54 Сьогодні UA [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.segodnya.ua/lifestyle/fun/pochti-kak-u-google-chemudivlyayut-ofisy-ukrainskih-it-kompaniy--764025.html> – відкритий.

55 Левон М.С. Система факторов, влияющих на производительность труда в it-фирмах // Экономика, Статистика и Информатика. – 2015. – №3. С.127-128.

56 Производительность труда на предприятии: измерение, анализ и повышение [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.gd.ru/articles/3575-proizvoditelnost-truda> – відкритий.

57 ДБН В.2.5.56:2014. Системи протипожежного захисту. Київ: Міністерство регіонального розвитку та будівництва України.

58 Желібо, Євген Петрович, and І. С. Сагайдак. "Безпека життєдіяльності." (2011).

59 Депутат, О. П., І. В. Коваленко, and І. С. Мужик. "Цивільна оборона. Підручник/За ред. Полковника ВС Франчука.–2-ге вид., доп." *Львів, Афіша* (2001).

60 Юрченко, Любов Іванівна. "Екологія." (2009).

61 Гавришев, Сергей Евгеньевич, et al. "Определение ценности техногенных георесурсов." *Вестник Магнитогорского государственного технического университета им. ГИ Носова* 2 (2010).

62 Потіш, А. Ф., et al. "Екологія: теоретичні основи і практикум." *Л.: Магнолія* (2006)..

63 Юрченко, Любов Іванівна. "Екологія." (2009).

64 Туниця, Т. Ю. "Збалансоване природокористування: національний і міжнародний контекст." *К.: Знання* 300 (2006).

65 Гавришев, Сергей Евгеньевич, et al. "Определение ценности техногенных георесурсов." *Вестник Магнитогорского государственного технического университета им. ГИ Носова* 2 (2010).

66 Коробкин, Владимир Иванович, and Леонид Васильевич Передельский. *Экология*. Феникс, 2010.

67 Виговська, Т. В. "Науково-популярний екологічний журнал «Екологічний вісник» № 2 (54)." *К.: Всеукраїнська екологічна ліга* (2009).